Expressive models in online learning

by Yogeshwer Sharma

EXPRESSIVE MODELS IN ONLINE LEARNING

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Yogeshwer Sharma

August 2010

EXPRESSIVE MODELS IN ONLINE LEARNING

Yogeshwer Sharma, Ph.D.

Cornell University 2010

We study the online learning model: a widely applicable model for making repeated choices in an interactive environment. In standard online learning model (or an online learning problem), the decision-maker is provided with a set of alternatives, and selects one alternative in each of the T sequential trials, deriving a reward for each selection. After T trials, the total reward of the decision-maker is compared with the best "single-arm" strategy which has the maximum reward in hindsight. The difference between the reward of the best single-arm strategy and that of the algorithm is called the *regret*, and one seeks decision-making algorithms whose regret is sublinear in T and running time is polynomial in the problem size.

In this thesis, we extend the basic online learning model in two important ways. In the first extension, we model sponsored search auctions as a multi-armed bandit problem (a type of online learning problem), and allow the alternatives (or advertisers in this case) to be *strategic* which can report possibly wrong rewards (in order to make personal gains). We seek to provide incentives to advertisers so as to get good solutions (socially efficient solutions). We prove that any socially efficient solution that provides right incentives to advertisers (being dominant strategy truthful) must suffer much higher regret than the regret suffered by algorithms for multi-armed problem without incentive issues.

In the second extension, also motivated by sponsored search and resource selection in distributed systems, we allow the set of available alternatives to vary over time, provide a natural way to define the regret, and give policies for the

decision-maker that suffer low regret. We also prove that the regret suffered by our policies is information-theoretically the lowest possible.

BIOGRAPHICAL SKETCH

Yogeshwer Sharma was born on September 14, 1983 in Gudha, a small village in the state of Haryana in India. After finishing his primary schooling in Gudha, he studied in a boarding school (Jawahar Navodaya Vidyalaya, Kareera) starting 6-th grade. For his undergraduate education, he went to IIT Kanpur, where he finished with a B. Tech. degree in Computer Science and Engineering in May 2004. He expects to graduate from Cornell University in August 2010, with a Ph. D. in Computer Science.

To my *mother*,

who is not here anymore in this physical existence;

To my *grandmother*,

who cared for me and never let me feel the void;

To my *father*,

for serving the need of a mother, and for being a truly wonderful father.

ACKNOWLEDGEMENTS

A time to stop and look back at how many wonderful people I have had in my life, who have made the life a wonderful adventure, inside the graduate school and outside of it.

First of all, I am indebted to my advisors Professor David P. Williamson and Professor Robert D. Kleinberg. It is hard to recount the many ways in which they have made this dissertation possible.

Thanks David for helping me learn the process of research and for supporting me through some rough spots and being there when I needed. I really appreciate the blend of freedom and structure I got while working with you. And of course, thanks for some wonderful dinners with your family and friends.

Thanks Bobby for your confidence in me, and your belief in my ability, even more so than I had for myself at times. I have learnt much from you through our interactions over the last few years. Your advice about various walks of life have been truly helpful for me, and I have been truly amazed at the breadth of advice I have been fortunate enough to get from you. Thank you for everything!

I would like to extend my thanks to my thesis committee members: David Shmoys and Ken Birman, for their time and support.

Thanks to my co-authors: Moshe Babaioff, Robert Kleinberg, Chandrashekhar Nagarajan, Alex Niculescu-Mizil, Alex Slivkins, Chaitanya Swamy, and David Williamson. You all have been very helpful in getting me out of graduate school! ☺

I have been fortunate enough to be a teaching assistant for Professor John Hopcroft a few times. Thanks John for all the teaching and career advice.

Cornell University, and specifically Computer Science department has provided a wonderful work place in which I have learnt a lot. I have been fortunate enough to

have spent last few years here. Thanks everybody in the department! And thanks Becky and Stephanie for always going that extra mile to make the administrative stuff so easy for students!

It has been great to meet some wonderful and understanding people in Ithaca. I am truly grateful for that. Ithaca is a wonderful town, and its people are very pleasant. I have so many sweet memories associated with this place. It has been really fun living here for last 6 years and I look forward to someday when I can come back and live here!

First of all, thanks to my roommates over last few years for sharing fun times: Vikram, Rony, Sameer, Malik, Yee Jiun (I see, it was not in Ithaca...), and Ankush. Thanks Yee Jiun for all the wonderful conversations, and thanks Ankush for introducing me to some truly caring and wonderful people like you in Ithaca. It has also been great to interact with Art of Living community and Bahà'ì community of Ithaca.

Thanks Lukas for being a wonderful, understanding, and genuinely caring friend. I have truly enjoyed those really long and spontaneous conversations about anything and everything. I will miss you and having those conversations in person.

Thanks Cristina for your love and understanding, and for letting me be who I am.

And finally, I have grown in so many different ways during my many years at Cornell. The journey has been rough and hard at various times, but it has been worth it throughout.

Thanks Bhai and Papa Ji for everything!

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

What should a decision maker do when faced with the problem of making a sequence of choices? For example, how should a search engine allocate its advertisement space to different competing advertisers? How should a doctor decide which treatments to administer to patients in order to ensure good results? How should a forecaster make predictions in order to be right most of the time? In this thesis, we focus on a mathematical model that formalizes these and related questions.

In learning theory literature, these types problems fall under the umbrella of *online learning*. This online learning framework is used for modelling and reasoning about the uncertainty in the environment, and has been of immense practical applicability in areas ranging from computer science and machine learning to statistics. In the section below, we give an informal introduction to the online learning framework, and then point to applications where this framework has been very fruitful.

1.1 An informal introduction to online learning

Before formalizing the online learning framework, let us understand what it hopes to capture. The basic idea behind online learning is that

1. the input to the problem is revealed over time to the algorithm, and
2. the algorithm (or the decision maker) makes choices in a sequence of time rounds in order to optimize some measure of its performance.

As is evident from the generality of the definition, this framework models a variety of processes. In the stock market, the information about the price of a stock becomes available over time, and the algorithmic trading agent tries to maximize its net gain over a sequence of days. In a weather forecasting service, the "expert advice" about the weather is revealed over time, and the decision maker tries to make good prediction based on its current information. In a peer to peer system, the information about network latencies is gathered by the algorithm over time, and the algorithm must make decisions to download files from a set of available hosts with this limited information. And the list continues.

To fix the terminology, we will call the process generating the input an *adversary* (denoted `adv`). In many applications, it is hard to model this process, and we assume either that the input is generated by some appropriate distribution, or in an arbitrary (worst-case) way. We will denote the decision-maker (or the algorithm) by `alg`. These are the two *competing entities* in the "game". At a high level, they interact for some number of rounds (generally denoted by T, which may or may not be fixed) in sequence. In each round, there are some number of *options* (or *alternatives* or *arms*) to choose from for the `alg`: the algorithm chooses one of the alternatives, and the adversary chooses the reward for each alternative in that round. At the end of the round, the adversary learns what alternative the algorithm chose, and the algorithm learns some feedback about the rewards in that round and how it did in that round (this feedback is specified by the feedback model, that we will formally define in a later section), and also collects the reward for its chosen alternative. The goal of the algorithm is to maximize the cumulative reward collected over the time horizon.

The attractiveness of the model comes from its simplicity and power. It is simple and natural, and at the same time very powerful in that it can model a variety of problems. Before we formalize the model and give precise definitions, we will point out the breadth of applications that can be modelled using the generic online learning framework described above.

1.1.1 Applications of online learning

Online learning has been very successful in modelling numerous application ranging from design of clinical trials to gambling. Due to its modelling capabilities, the framework has been a subject of intense study in theoretical computer science, machine learning, and statistics. We next present a small sample of the applications.

Design of clinical trials This is one of the original applications of the online learning problem (Berry and Pearson, 1985). In the design of clinical trials, the decision-maker is an experimenter (a doctor) who administers one of K experimental treatments to a sequence of patients, without knowing which treatment is the most effective. If the patient benefits from the treatment, we associate it with large "reward" for the decision maker, and vice-versa. The goal is to design a strategy for administering treatments so that the cumulative treatment of all patients is effective. By thinking of K treatments as the K options in the online learning problem, the problem is well-modelled as an online learning problem.

Algorithms in networks Imagine a (decision-maker) host in a peer-to-peer (P2P) network intending to download a sequence of files from other hosts in the system. When a file is downloaded from a particular host, the decision-

maker suffers a certain latency. How does the algorithm minimize the total latency suffered by the sequence of all downloads? By identifying the hosts with the arms, and negative latency as the rewards, this problem can be modelled as an online learning problem.

Also note that there is nothing special about "downloading files" in the above example. We can consider a host in a P2P network trying to "acquire a resource" or "find some information" over a sequence of rounds, and trying to maximize the overall quality. By associating a reward function with every host in each time round (negative latency, bandwidth etc.), we can model this general problem as an online learning problem. For instance, a DNS lookup algorithm can be modelled in this framework by identifying the DNS servers as arms, and negative latency of lookup time with the reward of a particular DNS server.

Ranking in search engine How should a search engine order the search results in order to give the most relevant results to the users? If we identify the "ranking of results" with arms, and "user's satisfaction with the ranking" with the reward of the ranking, the problem of choosing a good ranking can be modelled as an online learning problem (see (Radlinski et al., 2008)).

Online sponsored search auctions Which advertisements a search-engine should put in the ad-slots in order to get the most number of clicks and show ads with high value? Which ads should be put on webpages in order to get large number of clicks (Madani and DeCoste, 2005)? These problems can naturally be modelled as online learning problems by identifying ads with arms and value per click of an ad (or probability of getting a click) as the reward for choosing a particular arm.

These applications are only a few from a large number of applications of the online learning problems. Instead of surveying other such applications, we will now focus on precisely defining the *online learning problem*, which is the subject of this thesis.

1.2 Definition of online learning problems

Recall from the informal discussion above that in the online-learning framework, there are two "entities" called the algorithm (`alg`) and the adversary (`adv`). They interact for some number of indeterminate or fixed rounds, commonly denoted by T. In the beginning of the interaction, some information is exchanged between the algorithm and the adversary (depending on the particular problem). Then, in each round in turn,

- firstly, the adversary specifies the set of alternatives $K(t) \subseteq K$ the algorithm can choose from;
- secondly, the (`adv`) selects some reward function $r(t) : K(t) \to [0, 1]$ from a set Γ of all reward functions (this could be the set of all functions from $K(t)$ to $[0, 1]$) without revealing it to the `alg` and the algorithm (`alg`) selects one of the alternatives $i \in K(t)$ to play (without revealing it to the `adv`) simultaneously, and
- thirdly, the adversary learns which alternative $i \in K(t)$ was chosen, the algorithm learns some feedback about the rewards (examples of which will be given later), and the algorithm collects the reward $r_i(t)$ corresponding to alternative chosen (see Figure 1.1).

In the end, there is possibly some additional communication between the algorithm and the adversary. Both algorithm and adversary could be randomized.

The exact meaning of what *feedback* the algorithm learns depends on the particular online learning problem. For example, in the design of clinical trials, the doctor gets the feedback about how a particular treatment worked for a particular patient and not necessarily about how other treatments might have worked for the patient. For now, let us denoted the feedback model by $\mathfrak{F}$. This is a function that takes reward function r and the chosen arm i as an argument, and returns the "feedback", $\mathfrak{F}(r, i)$. The set of all feedbacks is denoted by $\texttt{range}(\mathfrak{F})$. The online learning problem is specified by $(K, \Gamma, \mathfrak{F})$, where Γ is the set of all reward functions chosen by the adversary.

In the next section, we give a formal definition of the online-learning problems, and of two particular instantiations thereof: the best-expert problem and the multi-armed bandit problem.

1.2.1 Arms, rewards, and feedbacks

An online learning problem specifies a set of alternatives $[K] := \{1, 2, \ldots, K\}$ which are the possible alternative to choose from, and a set of rounds $[T] := \{1, 2, \ldots, T\}$. The set of reward functions is denoted by Γ; each member of Γ is a function $r : [K] \to [0, 1]$ (we allow rewards from a bounded set, but we normalize the bounded set to $[0, 1]$ for ease of exposition). The reward of arm i at time t is denoted by $r_i(t)$, and the sum of rewards of arm i from time s to time t (i.e., $\sum_{u=s}^{t} r_i(u)$) is denoted by $r_i(s : t)$. The feedback received by the algorithm is determined by the "feedback function" $\mathfrak{F}$, which is a function $\mathfrak{F} : \Gamma \times [K] \to \texttt{range}(\mathfrak{F})$. The online problem is denoted by the tuple $(K, \Gamma, \mathfrak{F})$.

1 Some communication happens between algorithms and adversary in the very beginning.

2 **FOR** $t = 1, 2, \ldots, T$ (T may or may not be known in advance)

3 Adversary specifies the set of alternatives $K(t)$.

4 Simultaneously, (i) algorithm picks one of the alternatives, say $i(t)$, and (ii) adversary chooses rewards for each of the alternatives, say $r_j(t) \in [0, 1]$ for $j \in K(t)$.

5 Simultaneously, (i) the choice of algorithms revealed to the adversary, and (ii) some feedback about the rewards is revealed to the algorithm.

6 Algorithm collects reward for the alternative chosen, that is $r_{i(t)}(t)$.

7 Some additional communication happens between algorithm and adversary at the very end.

Figure 1.1: Interaction between adversary and algorithm in an online-learning problem.

We now discuss in detail the various ingredients of the model.

Alternatives or arms or options The set of alternatives is usually denoted by integers 1 through K, but could be any set in general, for example, a set of advertisements to choose from in an online sponsored search application.

Rewards We restrict the rewards of actions to be from $[0, 1]$. The rewards for options can be modelled in various ways depending on the application in discussion. We usually consider three such ways.

i.i.d. (or stochastic) rewards The reward of arm i is chosen according to a probability distribution P_i over $[0, 1]$. That is, the reward of arm i

in round t, $r_i(t)$, is a random sample from P_i (and similarly for other arms).

In this case, the *quality* of an arm can be measured by the mean of its reward distribution, that is, $\mathbb{E}_{X \sim P_i}[X]$. Typically, P_i is a Bernoulli distribution with support $\{0, 1\}$ with a fixed mean μ_i.

oblivious rewards In this case, the reward for all K options during all T rounds are decided in the beginning of the interaction, and cannot be changed thereafter. In particular, the rewards cannot depend on the algorithm's choices and behaviour (but don't have to be stochastic either).

adaptive rewards In this case, the rewards in round t are decided after interaction up to time $t-1$ has happened. The rewards could be arbitrary, and could depend on the history of play.

This is the most general model and applies when it is hard to model the quality of various options.

Types of feedback The feedback is a general function of reward function r and the option $i \in [K]$. This is the information the algorithm gets after round t about how it "performed" in that round. We consider two types of feedbacks: (i) full-feedback model, and (ii) partial-feedback model. In the *full-feedback model*, the algorithm is informed of the rewards of all alternatives in that round, giving rise to a special instantiation of the online learning problem called the *best-expert problem*. In the *partial-feedback model*, the algorithm is told only about the reward of the alternative it chose, and not about the rewards of alternatives it did not choose, giving rise to another important instantiation of the online learning problem called the *multi-armed bandit problem*. Let us discuss these two types of online learning problem.

1. In the best-expert problem (online learning problem with full feedback), the algorithm is given full information about the rewards in the previous time steps. In this case, if the adversary chooses reward function $r(t) \in \Gamma$ and the algorithm chooses alternative $i(t) \in K(t)$, the feedback $\mathfrak{F}(r(t), i(t))$ given to the algorithm is $r(t)$, which is a function $r(t) : K(t) \to [0, 1]$, i.e., the algorithm sees the rewards for all the possible choices it could have made. Note that the algorithm still collects the reward $r_i(t)$.

   A sample problem captured by this model, for instance, is that of aggregating information from K weather forecast websites to generate a new forecast. The algorithm collects a reward of 1 if its forecast is correct, and 0 otherwise. The goal is to maximize the cumulative reward. This problem falls in the full feedback model, because at the end of the day, the algorithm can observe for all weather forecasts if their prediction was correct or not.

2. In the multi-armed bandit problem (the online learning problem with partial feedback), the algorithm is given only the information about the reward of the alternative it chose (and not about the alternatives it did not choose). If the adversary chooses reward function $r(t)$ and the algorithm chooses alternative $i(t)$, the algorithm is given the feedback $\mathfrak{F}(r(t), i(t)) = r_{i(t)}(t)$.

   As an example, this feedback model captures the application like downloading a sequence of files in a P2P network, since it is in general not possible for the algorithm to know how long it would have taken to download a file from a different host than it chose.

1.2.2 Algorithm and adversary

An algorithm is a probability space $\Phi_{\texttt{alg}}$ with sample space $\Omega_{\texttt{alg}}$ (which is endowed with standard Borel measure, say) and a sequence of functions $\texttt{alg}(t) : \Omega_{\texttt{alg}} \times (\texttt{range}(\mathfrak{F}))^{t-1} \to K(t)$, for $t = 1, 2, \ldots$, where $K(t) \subseteq K$ is the set of available strategies (as will be explained when we describe the adversary). We interpret this as follows: if the random seed of the algorithm is $\omega_{\texttt{alg}} \in \Omega_{\texttt{alg}}$ and it has gotten the feedback $f(1), f(2), \ldots, f(t-1)$ in previous $t-1$ rounds, then it chooses the arm $(\texttt{alg}(t))(\omega_{\texttt{alg}}, f(1), f(2), \ldots, f(t-1))$ to be played in round t.

An adversary similarly consists of a probability space $\Phi_{\texttt{adv}}$ with sample space $\Omega_{\texttt{adv}}$ (which is endowed with standard Borel measure) and a sequence of functions $\texttt{adv}^+(t) : \Omega_{\texttt{adv}} \times K^{t-1} \to 2^{[K]}$ and $\texttt{adv}(t) : \Omega_{\texttt{adv}} \times K^{t-1} \to \Gamma$, for $t = 1, 2, \ldots$, where Γ is the set of reward functions. We interpret this as the following: when the adversary has the random seed $\omega_{\texttt{adv}}$ and the algorithm has chosen alternatives $i(1), i(2), \ldots, i(t-1)$ in the past, the adversary chooses $(\texttt{adv}^+(t))(\omega_{\texttt{adv}}, i(1), i(2), \ldots, i(t-1))$ as the set of available options/arms for round t and endows it with reward function $(\texttt{adv}(t))(\omega_{\texttt{adv}}, i(1), i(2), \ldots, i(t-1))$.

Depending on how the adversary chooses the rewards, the adversary can be i.i.d. (or stochastic), oblivious, or adaptive.

What this definition formalizes is the idea that the algorithm gets a sequence of coin flips in form of $\omega_{\texttt{alg}}$ (in case the algorithm is randomized), and in each round it has access to feedback about the decisions it has made up to this time. Given this information, it outputs a alternative to be chosen in the next time round. Similarly, the adversary could be randomized and gets a sequence of coin flips in form of $\omega_{\texttt{adv}}$. It gets to observe the alternatives chosen by the algorithm up to

time $t-1$ and chooses the set of available alternatives for round t and a reward function that assigns reward to the chosen (available) alternatives.

1.2.3 Objective function: Regret

As we mentioned earlier, the goal of the algorithm is to maximize the reward collected in the run of T rounds. We will measure the quality of the solution produced by an algorithm in terms of what is known as *regret*. Let us define the regret next.

First we define the T-step regret of an algorithm against a fixed adversary with respect to a fixed strategy x. A strategy is just a rule for picking arms in a sequence of rounds.

$$\texttt{regret}(\texttt{alg}, \texttt{adv}; x, T) = \mathbb{E}\left[\sum_{t=1}^{T} (r_x(t) - r_{\texttt{alg}}(t))\right].$$

Here, the expectation is taken with respect to the probability space generated by the random variables that `alg` and `adv` generate (that is, the probability measure generated by random variables $\{\texttt{alg}(t)\}_{t=1}^{T}$, $\{\texttt{adv}^{+}(t)\}_{t=1}^{T}$, and $\{\mathfrak{F}(\texttt{adv}(t), \texttt{alg}(t))\}_{t=1}^{T}$).

The strategy x to compare the regret against is usually the one of picking a fixed arm in every time round, but we will consider other strategies too. The strategy of always picking arm i will be denoted by $\texttt{single-arm}_i$. Note that we abuse notation by using $r_x(t)$ to denote the reward of strategy x at time t, even when x is not a fixed arm.

We next define the T-step regret of an algorithm `alg` against adversary `adv`

with respect to a *set of strategies* X.

$$\texttt{regret}(\texttt{alg}, \texttt{adv}; X, T) = \sup_{x \in X} \{\texttt{regret}(\texttt{alg}, \texttt{adv}; x, T)\}. \tag{1.2.1}$$

In the above definition, X is typically taken to be $\cup_{i \in K}\{\texttt{single-arm}_i\}$, in which case the regret is against picking the single best arm (or against the *single-best arm benchmark*). Similarly, the T-step regret of an algorithm against a family of adversaries $\texttt{ADV}$ with respect to a set of strategies can be defined as

$$\texttt{regret}(\texttt{alg}, \texttt{ADV}; X, T) = \sup_{\texttt{adv} \in \texttt{ADV}} \{\texttt{regret}(\texttt{alg}, \texttt{adv}; X, T)\}.$$

We also define the *strong regret* as

$$\widehat{\texttt{regret}}(\texttt{alg}, \texttt{adv}; X, T) = \mathbb{E}\left[\sup_{x \in X}\left(\sum_{t=1}^{T}(r_x(t) - r_{\texttt{alg}}(t))\right)\right].$$

Notice the difference between $\texttt{regret}(\texttt{alg}, \texttt{adv}; X, T)$ and $\widehat{\texttt{regret}}(\texttt{alg}, \texttt{adv}; X, T)$. In the former, it is the maximum of expectations; in latter, it is expectation of the maximums. The *(weak) regret* and *strong regret* are also called *ex-ante regret* and *ex-post regret* respectively. Using the fact that for random variables $X_1, X_2, \ldots, X_n$ defined over the same probability space, $\mathbb{E}[\max_i X_i] \geq \max_i \mathbb{E}[X_i]$, it follows that

$$\widehat{\texttt{regret}}(\texttt{alg}, \texttt{ADV}; X, T) \geq \texttt{regret}(\texttt{alg}, \texttt{ADV}; X, T).$$

This finishes the definition of regret. Let us now turn our attention to discussing why this is a good measure of quality of an algorithm.

Why is regret a good measure? When we measure the performance of an algorithm in terms of its regret with respect to the set $X = \cup_{i \in K}\{\texttt{single-arm}_i\}$, we are comparing the algorithm that is allowed to pick any allowable arm at any time with a benchmark which is *restricted* in that it can pick only one fixed arm in all rounds. Why is this a fair comparison and why is this a good measure?

It is easy to see that if the algorithm were allowed to *know* the distributions $P_1, P_2, \ldots, P_K$ (in i.i.d. adversary case), then the best any algorithm could do is to pick the alternative with the maximum expectation, i.e., alternative that achieves the argmax in $\arg\max_i \mathbb{E}_{X \sim P_i}[X]$, at least in the case when the set of allowable strategies is fixed to K. Now, if the algorithm is not allowed to know the distributions, then the regret measure how much the algorithm *loses* by not knowing the distributions. Or in other words, how much the algorithms should be "willing" to pay to know the distributions.

Remarks about maximization versus minimization There are usually two versions of online learning problems: the maximization version in which the goal is to maximize the rewards, and the minimization version in which the goal is to minimize the costs. In this thesis, we only consider the maximization version. To get the results for the minimization version, we can take the reward to be negative cost, and minimize the regret in the resulting maximization version.

Remarks about Baysian multi-armed bandits In this work, we don't make any assumptions about beliefs about the quality of different arms. In particular, there are no priors on how profitable (or bad) the arms are. In this sense, the problems we are considering are called *Worst Case Online Problems*.

There is a rich literature on the Bayesian version of the problem where the algorithm gets as input a prior on the reward distribution of each arm, and then is required to collect as much reward in expectation as possible (expectation is taken with respect to the randomness used by the algorithm as well as the randomness used in deciding the quality of the arms in the beginning). For this line of work (which is not considered in thesis), see (Gittins, 1979; Gittins and Jones, 1979; Gittins, 1989).

1.3 Problems considered in this thesis

In this thesis, we consider generalizations of the best-expert problem and the multi-armed bandit problem. The main thrust of our work is to analyze various generalizations of the basic online-learning framework (both the multi-armed bandit problem and best-expert problem), which are motivated by applications which we discuss next. Detailed discussion of related work appears in respective chapters.

We noticed earlier that the online learning framework models many applications, and is very widely applicable. Despite its applicability, there are certain limitations of the basic model described above. For example, it is implicit in the algorithms that the number of arms is small (this is not a hard restriction, but otherwise the regret becomes very large, since the regret grows polynomially with number of arms), all arms are always available to be picked (that is, $K(t)$ is restricted to be K in each round), their rewards can be observed accurately, the decision maker is risk-neutral (which is why regret is defined in terms of expectation of rewards) and so on.

The focus of thesis is to notice that for some applications of the multi-armed bandit (MAB) problem and the best-expert (BEX) problem these assumptions don't hold. Thus, we relax the assumptions in order to model the application scenarios more faithfully. The applications we consider come from sponsored search and computer networks. The two important scenarios that are considered are (1) online sponsored search auctions, where the arms are strategic advertisers, and the algorithm cannot observe the true rewards, and (2) many cases of interest in computer systems where not all options are always available.

We describe these two applications in turn.

1.3.1 Truthful MAB problem

As mentioned before, one important assumption in the MAB problem is that the algorithm obtains (and can observe) the *accurate* reward for the chosen arm (that is, the feedback is the actual reward of the chosen arm). Although this is often the case, there are many important exceptions. Consider, for example, sponsored search auctions, where a search engine needs to decide what advertisement (ad) to show in an ad slot alongside the search results, in order to maximize the social welfare. See Figure 1.3.1.

More concretely, the search engine (or the decision-maker) has the opportunity to show a few ads besides each search query. Let us assume that there is space for only one ad. Every time a shown ad is clicked (by a search-engine user), the total value (or reward) derived by the collective "system" (search-engine and advertisers) is a value specific to that ad (which is called the *value per click* for that ad). Also, when an ad is shown, it is clicked with an ad-specific probability called its *click-through rate*. Naturally, nobody knows the click-through rate of an ad (until it is shown a few times and its click-through rate estimated) and the value-per-click of an ad is *private information* for the owner of that ad. One goal of the search engine is to show ads in a sequence of rounds in order to maximize the overall value derived by the collective system (called the *social welfare*).

Notice that the task of maximizing the social welfare is made difficult by the fact that when a specific ad is shown and clicked on, the search engine doesn't know precisely how much "reward" it got, since the reward derived from the click is equal to the value-per-click of the ad which is a private information of the owner of the ad.

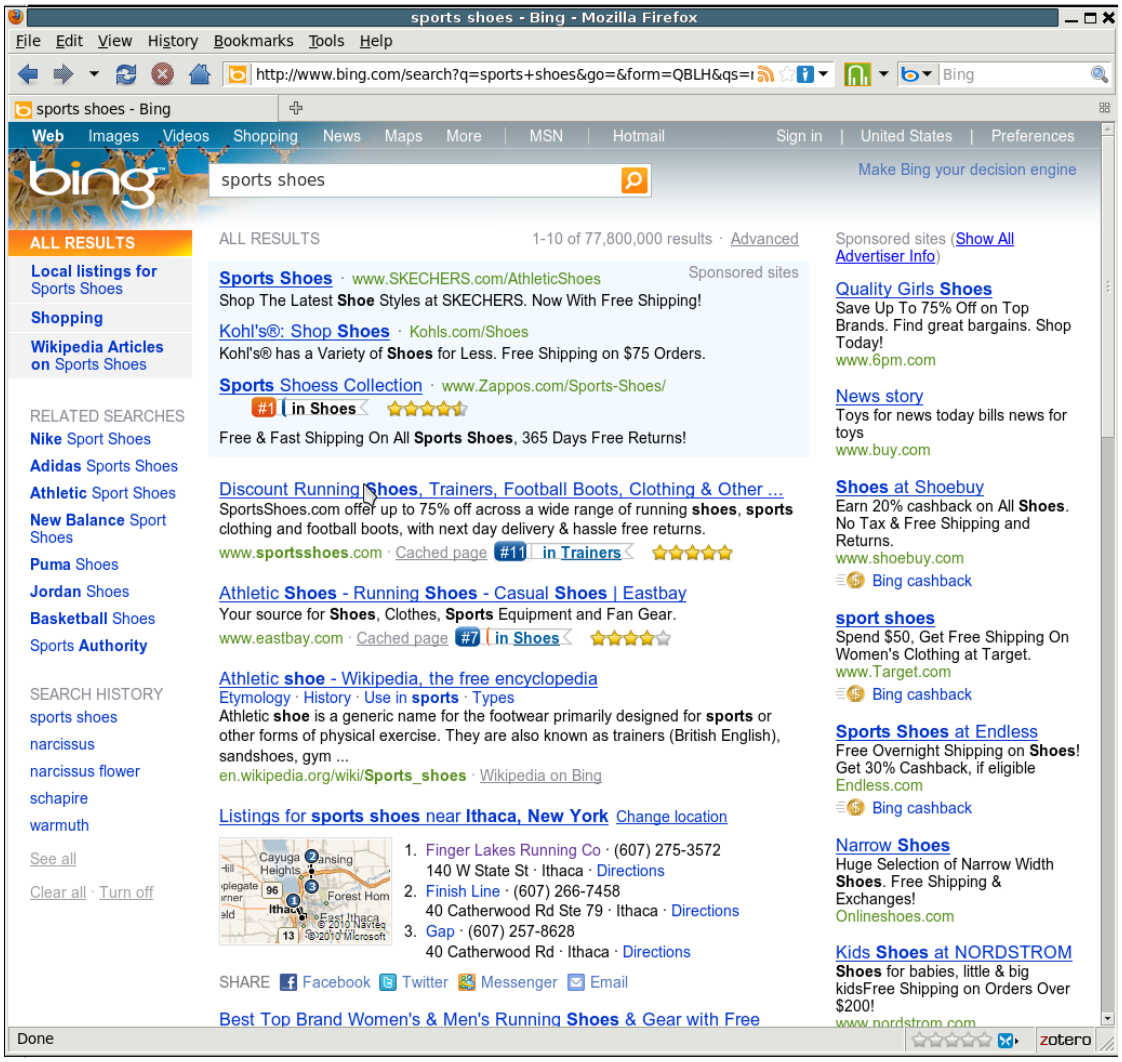

Figure 1.2: Illustration of sponsored search. In response to the user query ("sports shoes" in this case), the search engine shows a set of search results as well as some advertisements (sponsored links) on top (light blue background in this figure) and right side.

When we model sponsored search auction as an MAB problem by identifying ad i with arm i that yields expected reward $c_i v_i$ (c_i: *click through rate*, v_i: *value per click*), the arms have effectively become strategic agents, because v_i is the value that ad i derives when clicked on, which only ad i knows about and the algorithm cannot directly observe it. If the search-engine asks the ads for their value-per-click, they might have an incentive to either understate or overstate their value per click. We therefore need to look into incentivizing the strategic ads to reveal their true value-per-click. We focus on the solution/equilibrium where every ad gets the most benefit by telling the true value per click to the search engine. How do we design the algorithm such that truthtelling is indeed an equilibrium?

In Chapter 3, we consider the *truthful multi-armed bandit* problem: a strategic version of the classical MAB problem which models the sponsored search auction scenario described above. We model the auction as a mechanism design problem, in which each agent (ad) i bids a value b_i as a proxy for its true value per click v_i (b_i may not be equal to v_i). The allocation algorithm then allocates ads (to the *single* ad slot) for T time rounds. At the end, it charges payments p_i from ad i (or from the owner of ad i), in return of showing its ad (called the pricing rule). The pair (allocation algorithm, pricing rule) is called the mechanism. It is *truthful* if each agent derives as much utility from bidding her true value as she can derive from bidding any other value, where the utility of agent i is defined as the number of clicks she got times her value per click minus the payment she paid. We aim to find a truthful mechanism that minimizes the regret (defined as in the classical MAB problem).

Can we find mechanisms that achieve regret for the truthful MAB problem that is as low as can be obtained for the classical MAB problem? We investigate this

question in detail in Chapter 3, and prove rigorously that the structure of truthful mechanisms for truthful MAB problem is very restrictive; in particular, they cannot simultaneously explore arms and exploit an empirically good arm. Here, "exploration" means choosing arms according to some predetermined distribution, independent of their perceived quality, and "exploitation" means choosing an arm that has been good in the past. At a high level, we prove the following result.

Main Theorem 1.3.1. *Let $\mathcal{A}$ be an allocation rule for truthful MAB problem which satisfies some natural and technical conditions*[1]*. This allocation rule can be made into a truthful mechanism by a pricing rule if and only if*

1. *$\mathcal{A}$ allocates an ad more often if the ad increases its bid, everything else remaining constant, and*
2. *If the allocation decision in a round depends on bids, then the feedback from that round (in form of a click/no-click) is not used later in the algorithm.*

Note that the first condition above is a form of monotonicity condition that is common to many truthful mechanisms (Archer and Tardos, 2001; Myerson, 1981; Nisan et al., 2007). The second condition, which can be viewed as separation between exploration and exploitation (see Chapter 3 for precise definitions), is novel to our analysis. It says that if the algorithm is exploiting a good arm in a particular round (by letting allocation decision depend on the bids), then it cannot simultaneously explore in the sense that the feedback from this round must not be used later in making allocation decisions.

As a result of the above result, truthful mechanisms end up performing much worse (sometimes exponentially worse) than the algorithms for the classical MAB

[1]See Theorem 3.2.3 for details.

problem. More quantitatively, any truthful mechanism must suffer a regret of $\Omega(K^{1/3}T^{2/3})$, while best algorithms for classical MAB problem are known to achieve regret $\tilde{O}(K^{1/2}T^{1/2})$, where K is the number of arms. This work also suggests that a truthful mechanism sometimes must show ads without charging anything, which is somewhat counterintuitive.

1.3.2 Sleeping MAB problem

Another implicit assumption in the MAB problem is the availability of arms in every round. Consider the problem of picking a good node from which to download a file in a peer-to-peer network (in general, the problem of using a resource in a distributed system). We can model this as an MAB problem by identifying nodes with arms. However, the assumption that arms are always available is clearly violated. In a realistic setting, nodes can enter and leave constantly, or can be down for maintenance, or certain parts of the network can be unreachable; thus there is no guarantee that any node is always available. In Chapter 4, we consider the *sleeping bandit and expert problems* where the set of actions that are available to the decision algorithm varies over time, and arms are arbitrarily unavailable in time rounds. With a few notable exceptions, such problems have been largely unaddressed in the literature. Departing from previous work on this "Sleeping Experts" problem, we compare algorithms against the payoff obtained by the *best ordering* of the actions, which is a natural benchmark for this type of problem. As no single arm is always available anymore, we suggest an extension of the "single-best arm benchmark" (as defined in the definition of regret, (1.2.1)), called the *ordering benchmark*, which orders the K arms according to the best possible order (out of $K!$ feasible orders), and always picks the first available arm in the chosen order. The ordering benchmark enjoys the following nice properties: (a) it reduces

to the usual single-best arm benchmark when all arms are available, (b) it is natural since this is how people seem to pick one option from the set of available options, and (c) it gives the optimal strategy when the algorithm knows the distribution of arms' rewards.

We study both the full-feedback (best expert) and partial-feedback (multi-armed bandit) settings and consider both stochastic and adversarial reward models. For all settings we give algorithms achieving (almost) information-theoretically optimal regret bounds (up to a constant or a sub-logarithmic factor) with respect to the best-ordering benchmark.

When the rewards for arms in each round are independent samples from arm-dependent distributions (i.e., when the adversary is i.i.d.), we provide an efficient algorithm which suffers a regret that is within a constant factor of what is achievable, both in full-feedback and partial-feedback settings. The lower bound is a particular novelty in this work, since it holds uniformly over time, rather than only in the limit of time horizon approaching infinity (Lai and Robbins, 1985a). The variant with adversarial rewards turns out to be more difficult. Although we provide algorithms that achieve regret within constant factor of optimal in the full-feedback setting and within sub-logarithmic factor of optimal in the partial-feedback setting, both algorithms have exponential running time. To appreciate the hardness of the fully adversarial case, we prove that, unless RP = NP, any low regret algorithm that learns internally a consistent ordering (see Theorem 4.3.3) over experts cannot be computationally efficient. Learning a consistent ordering just means that the algorithm chooses an arm by first choosing an ordering and then choosing the first available arm in the chosen ordering. Note that this does not mean that there can be no computationally efficient, low regret algorithms

for the fully adversarial case. There might exist learning algorithms that are able to achieve low regret without actually learning a consistent ordering over experts. Finding such algorithms, if they do indeed exist, remains an open problem.

1.4 Bibliographic notes

The material in Chapter 3 is joint work with Moshe Babaioff and Alex Slivkins, which appeared in (Babaioff et al., 2009). The material in Chapter 4 is joint work with Robert Kleinberg and Alex Niculescu-Mizil, which will appear in (Kleinberg et al., 2010).

CHAPTER 2

ALGORITHMS FROM LEARNING THEORY

In this chapter, we review some algorithms for online learning problems that will be used later in this thesis. We consider the following four problems:

1. Best expert problem with adversarial rewards: The algorithms for this problem are some of the first to appear for online learning problems (Littlestone and Warmuth, 1994; Freund and Schapire, 1997). We will use the presented algorithm in Section 4.3.1 to derive algorithm for sleeping version of the best expert problem.

2. Linear optimization problem with full-feedback: We haven't discussed this problem before. In this problems, the set of options/arms is continuous, let us say a subset of $\mathbb{R}^k$ (instead of discrete K arms), and the reward functions is restricted to be linear. We present this problem because the best expert problem can be reduced to linear optimization, and algorithms for linear optimization problems are easier to generalize to other setting we consider later in this thesis (see Section 4.2.1).

3. Stochastic multi-armed bandit problem: In this version of the multi-armed bandit problem, rewards are generated according to a predetermined distribution, and the algorithm learns the value of the random rewards for the arms it chooses. This algorithm will be generalized to the sleeping version of stochastic multi-armed bandit problem in Section 4.2.2.

4. Adversarial multi-armed bandit problem: In this version of the multi-armed bandit problem, there are no assumptions on the process generating the rewards. We also consider a slight variant of the problem (as detailed in

Section 2.4) from Auer et al. (2002a), which will be used in Chapter 4 (see Section 4.3.2).

Now, we turn out attention to the Hedge algorithm for the adversarial best-expert problem.

2.1 The Hedge algorithm

Let us recall the main ingredients of the adversarial best-expert setting from previous chapter.

- Options: K options/experts.
- Rewards: adversarial rewards with range $[0, 1]$.
- Feedback: full-feedback model.

The goal of the algorithm, as usual, is to maximize its reward. In this section, we present the Hedge algorithm (Freund and Schapire, 1997) which achieves a regret of $\mathcal{O}(\sqrt{T \log K})$ for this setting.

An informal description The algorithm starts at round $t = 1$, with weight vector $w_i(0) = 1$ for all $i \in K$. In the beginning of a typical round t, the algorithm has $w_i(t-1)$ for all $i \in [K]$. It selects one of the alternatives (each with probability proportional to the current weight of the alternative), observes rewards for *all* alternatives, and update all $w_i(t-1)$ to $w_i(t)$ for all $i \in [K]$ (according to an exponential update rule). The algorithm is presented in Figure 2.1.

The algorithm is randomized, and the adversary could be randomized too. The sigma-field generated by all the information algorithm has observed by time t is

PARAMETER: $\epsilon > 0$.

$w_i(0) = 1$ for all $i \in [K]$.

$t = 1$.

WHILE $t \leq T$

    $w(t-1) = \sum_{i \in [K]} w_i(t-1)$.

    $p_i(t) = w_i(t-1)/w(t-1)$.

    Select expert i with probability $p_i(t)$, that is $\texttt{alg}(t) = i$ with probability $p_i(t)$.

    // observe rewards $r_i(t)$ for all $i \in [K]$.

    update $w_i(t) = w_i(t-1)(1+\epsilon)^{r_i(t)}$.

    $t = t + 1$

Figure 2.1: The Hedge algorithm. The update rule $w_i(t) = w_i(t-1) \cdot e^{\gamma \cdot r_i(t)}$ can be analyzed in a somewhat different way, but essentially gives the same regret bound.

denoted by $F(t)$. Similarly, the adversary observes $\texttt{alg}(t)$, the expert chosen by algorithm in time t. The sigma-field generated by all the information observed by the adversary by time t is called $G(t)$. The sigma-field generated by all the information observed by both the algorithm and the adversary by time t (that is the union of $F(t)$ and $G(t)$) is denoted by $FG(t)$.

F and this

Analysis of algorithm In this section, we analyse the Hedge algorithm presented in Figure 2.1. The regret bound for the algorithm is presented in the following theorem.

Theorem 2.1.1. *The regret of the Hedge algorithm can be bounded by*

$$\mathbb{E}[r_{\max}(1:T)] - \mathbb{E}[r_{\texttt{alg}}(1:T)] \leq \epsilon \mathbb{E}[r_{\max}(1:T)] + \frac{\log K}{\epsilon}.$$

Here, and elsewhere, we will use $r_{\max}(\cdot)$ to stand for $\max_i r_i(\cdot)$. So, the regret is with respect to the best arm in hindsight.

We will also use $\mathbb{V}[\cdot]$ to denote the value of a random variable.

Proof. Note that the random variable $\mathbb{V}[w_i(t) \mid F(t)]$ is a constant function, and so is $\mathbb{V}[w_i(t) \mid FG(t)]$.

We will analyze how the weights grow as a function of time.

$$\mathbb{V}[w_i(t+1) \mid FG(t)] = w_i(t)(1+\epsilon)^{\mathbb{V}[r_i(t+1)|FG(t)]}, \quad \text{for } i \in [K].$$

Now, sum over all i to get

$$\begin{aligned}
\mathbb{V}[w(t+1) \mid FG(t)] &= \sum_i w_i(t)(1+\epsilon)^{\mathbb{V}[r_i(t+1)|FG(t)]} \\
&\leq \sum_i w_i(t)(1+\epsilon\mathbb{V}[r_i(t+1) \mid FG(t)]) \\
&= w(t) + \epsilon \sum_i w_i(t)\mathbb{V}[r_i(t+1) \mid FG(t)].
\end{aligned}$$

Divide both sides by $w(t)$ to get

$$\begin{aligned}
\frac{\mathbb{V}[w(t+1) \mid FG(t)]}{w(t)} &\leq 1 + \epsilon \sum_i \mathbb{V}[p_i(t+1) \mid FG(t)]\mathbb{V}[r_i(t+1) \mid FG(t)] \\
&= 1 + \epsilon \sum_i \mathbb{V}[p_i(t+1)r_i(t+1) \mid FG(t)] \quad \text{(because } p_i(t+1) \text{ is} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{constant given } FG(t).)
\end{aligned}$$

Taking expectation on both sides,

$$\frac{\mathbb{E}[w(t+1) \mid FG(t)]}{w(t)} \leq 1 + \epsilon\mathbb{E}[r_{\mathtt{alg}}(t+1) \mid FG(t)].$$

Note that before taking the expectation, the random variables were measurable with respect to $FG(t+1)$, and the expectation is taken so as to make the resulting expectation measurable with respect to $FG(t)$.

Take the log of both sides to get:

$$\log \mathbb{E}[w(t+1) \mid FG(t)] - \log w(t) \leq \epsilon \mathbb{E}[r_{\texttt{alg}}(t+1) \mid FG(t)].$$

Since $\mathbb{E}[\log(X)] \leq \log \mathbb{E}[X]$ we get

$$\mathbb{E}[\log w(t+1) - \log w(t) \mid FG(t)] \leq \epsilon \mathbb{E}[r_{\texttt{alg}}(t+1) \mid FG(t)].$$

Taking one more expectation yields the unconditional inequality.

$$\mathbb{E}[\log w(t+1) - \log w(t)] \leq \epsilon \mathbb{E}[r_{\texttt{alg}}(t+1)].$$

This inequality holds for $t = 0, 1, 2, \ldots, T-1$. Summing for all these values of t yields:

$$\mathbb{E}[\log w(T) - \log w(0)] \leq \epsilon \mathbb{E}[r_{\texttt{alg}}(1:T)].$$

Note that $w(0) = K$. Rearranging the terms we get

$$\mathbb{E}[r_{\texttt{alg}}(1:T)] \geq \frac{\mathbb{E}[\log w(T)]}{\epsilon} - \frac{\log K}{\epsilon}.$$

Let us now put a lower bound on $\mathbb{E}[\log w(T)]$. We have $\mathbb{V}[w(T) \mid FG(T)] \geq \mathbb{V}[w_i(T) \mid FG(T)] = \mathbb{V}[(1+\epsilon)^{r_i(1:T)} \mid FG(T)]$ for all $i \in [K]$. Taking the log shows that $\mathbb{V}[\log w(T) \mid FG(T)] \geq \mathbb{V}[r_i(1:T)\log(1+\epsilon) \mid FG(T)]$ for all i. Taking one more expectation yields the unconditional inequality $\mathbb{E}[\log w(T)] \geq \mathbb{E}[r_i(1:T)\log(1+\epsilon)]$ for all i. Using the inequality above, using the fact that $\frac{\log(1+\epsilon)}{\epsilon} \geq \frac{1}{1+\epsilon} \geq 1-\epsilon$, and rearranging the terms, we get

$$\mathbb{E}[r_i(1:T)] - \mathbb{E}[r_{\texttt{alg}}(1:T)] \geq \epsilon \mathbb{E}[r_i(1:T)] + \frac{\log K}{\epsilon}, \quad \text{for all } i.$$

Taking i as an expert in $\arg\max_j \mathbb{E}[r_j(1:t)]$ gives the result in the theorem. □

We can put an upper bound T on $\mathbb{E}[r_i(1:T)]$ for all i. Taking $\epsilon = \sqrt{(\log K)/T}$ gives a regret bound of $\mathcal{O}(\sqrt{T \log K})$.

2.2 Follow the leader algorithm

Follow the Leader algorithm (Hannan, 1957; Kalai and Vempala, 2005) is an algorithm for linear optimization, and not just for the expert problems we are considering, but it can be used to solve best-expert problem. Let us first outline the linear optimization problem, and then present the algorithm.

The linear optimization problem In the linear optimization problem, the set of strategies S is a compact set in $\mathbb{R}^d$ on which a linear function can be optimized. (To model the best-expert problem with K experts as a linear optimization problem, we take S to be convex hull of K unit vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K$.) The set of reward function Γ is the set of linear functions. Note that the reward functions can be identified with vectors in $\mathbb{R}^d$. For a reward function $r \in \mathbb{R}^d$, and strategy $x \in S$, the reward r_x is $r \cdot x$, where $(\cdot)$ denotes the inner product of vectors. The rewards are oblivious. The feedback model is the full-feedback model, which means that the algorithm observes the vector r at the end of the round. Here is a quick summary:

- Set of options: A compact set $S \subseteq \mathbb{R}^d$.
- Rewards: Linear functions over S. Oblivious to the choices of algorithm.
- Feedback model: full-feedback.

The "follow the leader" algorithm from Hannan (1957); Kalai and Vempala (2005) is shown in Figure 2.2.

Analysis of the algorithm Let us set up some notation that would be useful in the analysis of the algorithm. We denote by $r(i:j)$ the vector $\sum_{t=i}^{j} r(t)$, and by $\mathrm{OPT}(r)$ the vector $\arg\max_{x \in S} r \cdot x$. Therefore, according to our notation, $r_{\mathrm{OPT}(s)}(t)$

PARAMETER:

ϵ: length of the initial vector r_0 (see below).

Choose a random vector r_0 from $[-\frac{1}{\epsilon}, \frac{1}{\epsilon}]^d$.

FOR $t = 1, 2, \ldots, T$

Choose the strategy $\arg\max_{x \in S} \left(\sum_{s=0}^{t-1} r_s \cdot x\right)$

Figure 2.2: Follow the leader algorithm.

denotes $r(t) \cdot \mathrm{OPT}(s)$, that is the reward of strategy $k := \mathrm{OPT}(s)$ (that is optimal for reward function s) when the actual reward function is $r(t)$.

In the discussion below, p-norm (for $p \geq 1$) of a vector $x \in \mathbb{R}^d$ is $\|x\|_p := \left(\sum_{i=1}^d |x_i|^p\right)^{1/p}$, and p-norm of a set S is $\|S\|_p := \sup_{x,y \in S} \|x - y\|_p$. As a notational convenience, we denote $\lim_{p \to \infty} \|x\|_p$ by $\|x\|_\infty$. Taking the limit $p \to \infty$, $\|x\|_\infty := \max_{i=1,2,\ldots,d} |x_i - y_i|$, and $\|S\|_\infty = \sup_{x,y \in S} \|x - y\|_\infty$.

Theorem 2.2.1. *For any sequence of $r(1), r(2), \ldots, r(T)$, any $x \in S$, and for any $\epsilon > 0$,*

$$\sum_{t=1}^{T} r_{\mathrm{OPT}(r(0:t-1))}(t) \geq \sum_{t=1}^{T} r_x(t) - \frac{2}{\epsilon}\|S\|_1 - \sum_{t=1}^{T} \frac{\epsilon}{2}\|r(t)\|_1^2 \|S\|_1.$$

From this theorem, we can derive a regret bound for the best-expert problem also. The regret is at most a

$$\frac{2}{\epsilon}\|S\|_1 + \sum_{t=1}^{T} \frac{\epsilon}{2}\|r(t)\|_1^2 \|S\|_1.$$

In the best expert problem, S is the simplex in K dimensions ($\{x \in \mathbb{R}^K : \|x\|_1 = 1\}$), whose L_1 diameter is a constant. Since all rewards are bounded between 0 and 1, the vector of rewards $r(t)$ has L_1 norm at most K. Using $\epsilon = K^{-1}T^{-1/2}$, the "follow the leader" algorithm gives a regret bound of $\mathcal{O}(\sqrt{TK^2})$, which is slightly worse than the $\mathcal{O}(\sqrt{T \log K})$ bound of Hedge algorithm.

The idea of the analysis of the algorithm is to proceed in two parts.

1. Show that playing the "best cumulative strategy" for every time round (that is, $\mathrm{OPT}(r(0:t))$ in round t) is at least as good as playing any other fixed strategy (if we can play it).
2. Show that playing the "best cumulative strategy for past" for every time round (that is, $\mathrm{OPT}(r(0:t-1))$ in round t) is not much worse compared to the "best cumulative strategy" in the previous bullet point.

In the next two lemmas, we prove these results.

Lemma 2.2.2. *For every time t, and for every $x \in S$,*

$$r_x(0:t) := \sum_{s=0}^{t} r_x(s) \leq \sum_{s=0}^{t} r_{\mathrm{OPT}(r(0:t))}(t)$$

Proof. This can be proved easily by induction. For $t = 0$, the statement is true from the definition of $\mathrm{OPT}(r(0))$. Let us assume that the statement holds for $t-1$. For t, we have

$$\begin{aligned}
\sum_{s=0}^{t} r_{\mathrm{OPT}(r(0:s))}(s) &= \sum_{s=0}^{t-1} r_{\mathrm{OPT}(r(0:s))}(s) + r_{\mathrm{OPT}(r(0:t))}(t) \\
&\geq \sum_{s=0}^{t-1} r_{\mathrm{OPT}(r(0:t))}(s) + r_{\mathrm{OPT}(r(0:t))}(t) \quad \text{(From the induction hypothesis.)} \\
&= \sum_{s=0}^{t} r_{\mathrm{OPT}(r(0:t))}(s) \\
&\geq \sum_{s=0}^{t} r_x(s) \quad \text{for any } x \in S. \quad \text{(From the definition of } \mathrm{OPT}(r(0:t))\text{.)}
\end{aligned}$$

This proves the lemma. □

Once we know that playing $\mathrm{OPT}(r(0:t))$ at time t is as good as playing any other fixed strategy, why don't we just play it? The problem is that we don't know $r(t)$ before playing in round t. So, we now show that playing $\mathrm{OPT}(r(0:t-1))$ is not much worse.

Lemma 2.2.3. *For any t, we have*

$$\mathbb{E}[r_{\mathrm{OPT}(r(0:t-1))}(t)] \geq \mathbb{E}[r_{\mathrm{OPT}(r(0:t))}(t)] - \frac{\epsilon}{2}\|r(t)\|_1^2\|S\|_\infty,$$

where the expectation is taken over the randomness of the algorithm and that of the (oblivious) adversary.

Proof. Let us define $p := r(0:t-1)$ and $q := r(0:t)$, hence $q = p + r(t)$. (Note that both of these are random variables because $r(0)$ is random.) We will produce a random variable $\tilde{q}$ such that its distribution is the same as that of q, but it is coupled with p, such that $\mathbb{P}[p \neq \tilde{q}] \leq \frac{\epsilon}{2}\|r(t)\|_1$. Let us define

$$\tilde{r}(0) := \begin{cases} r(0) - r(t) & \text{if } r(0) - r(t) \in [-\frac{1}{\epsilon}, \frac{1}{\epsilon}]^d, \\ -r(0) & \text{otherwise.} \end{cases}$$

Now we define $\tilde{q} = \tilde{r}(0) + r(1:t)$. See Figure 2.2 for an illustration. Note that $\tilde{r}(0)$ has the same distribution as $r(0)$, and hence $\tilde{q}$ has the same distribution as q. Also, the probability that $\tilde{q}$ is not equal to p is equal to the volume of hatched regions in Figure 2.2, which can be bounded by

$$\begin{aligned} \mathbb{P}[p \neq \tilde{q}] &\leq \sum_{i=1}^{d} \left(\frac{2}{\epsilon}\right)^{d-1} |r_i(t)| \Big/ \left(\frac{2}{\epsilon}\right)^{d} \\ &= \frac{\epsilon}{2}\|r(t)\|_1. \end{aligned}$$

With this probability, when p and $\tilde{q}$ are different, the difference in $r_{\mathrm{OPT}(r(0:t-1))}(t)$ and $r_{\mathrm{OPT}(r(0:t))}(t)$ can be at most

$$\max_{x,y \in S} |r_x(t) - r_y(t)| = \max_{x,y} |r(t) \cdot (x - y)|$$

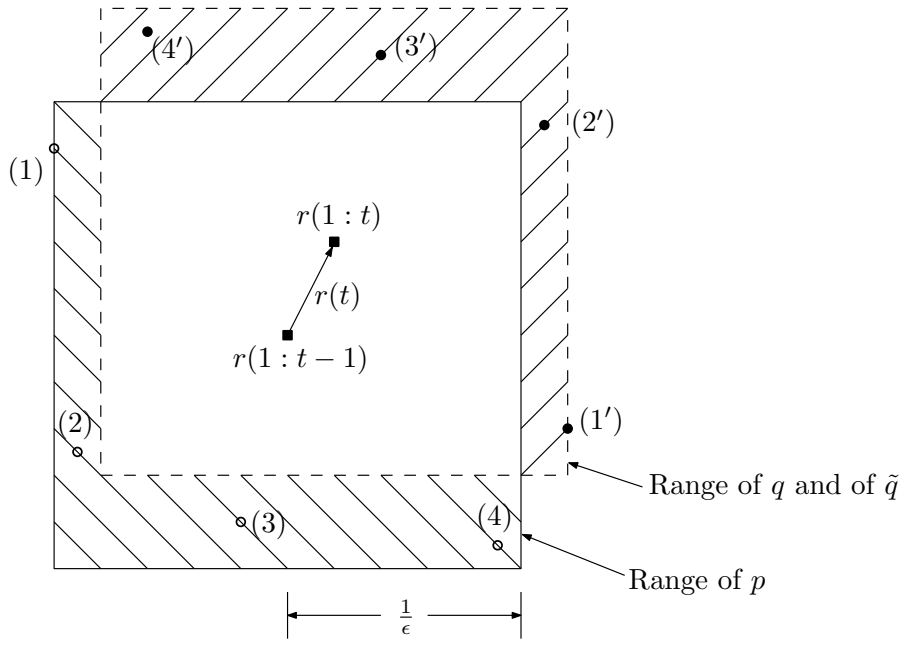

Figure 2.3: Illustration of coupling in Follow-The-Leader algorithm. The range of the random variable p (that is $r(1:t-1)$ added with random variable $r(0)$ from $[-\frac{1}{\epsilon}, \frac{1}{\epsilon}]^d$) is denoted by the solid square, and those of q and $\tilde{q}$ are denoted by the dashed square. After coupling of p and $\tilde{q}$, if p lies in the intersection of solid and dashed box, then so does $\tilde{q}$ (and they are equal). If p lies in the hatched region of the solid box (say it is equal to point labelled (i)), then $\tilde{q}$ lies in the hatched region of the dashed box (and it is equal to the point labelled (i')). The distribution of p is same as the distribution of $\tilde{q}$.

$$\leq \|r(t)\|_1 \max_{x,y \in S} \|x - y\|_\infty$$

$$= \|r(t)\|_1 \|S\|_\infty$$

Therefore, the difference in $\mathbb{E}[r_{\mathrm{OPT}(r(0:t-1))}(t)]$ and $\mathbb{E}[r_{\mathrm{OPT}(r(0:t))}(t)]$ can be at most $\frac{\epsilon}{2}\|r(t)\|_1\|r(t)\|_1\|S\|_\infty$, proving the lemma. □

Now, we are ready to prove the performance guarantee of the algorithm.

Proof of Theorem 2.2.1. We analyze the sum $\sum_{t=1}^{T} r_{\mathrm{OPT}(r(0:t-1))}(t)$. From

Lemma 2.2.3, we have

$$\sum_{t=1}^{T} r_{\mathrm{OPT}(r(0:t-1))}(t) \geq \sum_{t=1}^{T} r_{\mathrm{OPT}(r(0:t))}(t) - \sum_{t=1}^{T} \frac{\epsilon}{2}\|r(t)\|_1^2 \|S\|_\infty.$$

Let us focus on the first term on the right hand side. The strategy x is a "free variable" in these equations: the equations hold for any $x \in S$.

$$\begin{aligned}
\sum_{t=1}^{T} r_{\mathrm{OPT}(r(0:t))}(t) &= \sum_{t=0}^{T} r_{\mathrm{OPT}(r(0:t))}(t) - r_{\mathrm{OPT}(r(0))}(0) \\
&\geq \sum_{t=0}^{T} r_x(t) - r_{\mathrm{OPT}(r_0)}(0) \\
&\geq \sum_{t=1}^{T} r_x(t) - |r(0) \cdot (x - \mathrm{OPT}(r_0))| \\
&\geq \sum_{t=1}^{T} r_x(t) - \|r(0)\|_\infty \|S\|_1 \\
&= \sum_{t=1}^{T} r_x(t) - \frac{2}{\epsilon}\|S\|_1.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\sum_{t=1}^{T} r_{\mathrm{OPT}(r(0:t-1))}(t) &\geq \sum_{t=1}^{T} r_x(t) - \frac{2}{\epsilon}\|S\|_1 - \sum_{t=1}^{T} \frac{\epsilon}{2}\|r(t)\|_1^2 \|S\|_\infty \\
&\geq \sum_{t=1}^{T} r_x(t) - \frac{2}{\epsilon}\|S\|_1 - \sum_{t=1}^{T} \frac{\epsilon}{2}\|r(t)\|_1^2 \|S\|_1,
\end{aligned}$$

finishing the proof of the theorem. □

Note that we assume in the analysis that the adversary is oblivious (not adaptive).

2.3 UCB1 algorithm

In this section, we provide the UCB1 algorithm (Auer et al., 2002a) for the multi-armed bandit problem with i.i.d. adversary. Recall that there are K arms; let

μ_i denote the mean of arm i for $i = 1, 2, \ldots, K$. Here are the ingredients of the problem:

- Arms: K arms, numbered 1 through K.
- Rewards: Stochastic rewards for arm i that are drawn independently from distribution P_i with (fixed but unknown) mean μ_i.
- Feedback: partial-feedback model.

The idea of algorithm is to keep an *observed mean* for each arm, and also a *confidence interval* around each arm indicating how confident the algorithm is that the actual mean is within the confidence interval of the observed mean. Then the algorithm plays the most *optimistic arm*, assuming the actual mean of the arm might be at the highest point in the confidence interval. The algorithm is presented in Figure 2.4.

Analysis of UCB1 The following theorem bounds the regret of the UCB1 algorithm. Let arms be indexed by decreasing order of means, that is $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. We denote by Δ_i the difference between the mean of best arm and that of arm i, that is $\Delta_i := \max_j \mu_j - \mu_i$, which is equal to $\mu_1 - \mu_i$ if arms are ordered in decreasing means order.

Theorem 2.3.1. *The regret of* UCB1 *against an i.i.d. adversary can be bound by*

$$T \cdot (\max_i \mu_i) - \mathbb{E}[r_{\mathsf{UCB1}}(1:T)] \leq \mathcal{O}(KT^{-2}) + \sum_{i=2}^{K} \frac{32 \log T}{\Delta_i}.$$

Proof. Let us set up some notation first that will be useful in the analysis. We denote the value of variable v at time by $v(t)$. So, $n_i(t)$, for example, denotes the number of times arm i has been played up to time t.

VARIABLE:

t: current time round.

n_i: number of times arm i is played

R_i: total reward of arm i.

c_i: confidence interval for arm i

$t = 1$

WHILE $t \leq K$

play arm t at time t, and observe reward r_t

$n_t = 1$

$R_t = r_t$

$t = t + 1$

WHILE true

let $j := x(t) = \arg\max_{i \in K} \left(\frac{R_i}{n_i} + \sqrt{\frac{8 \log t}{n_i}} \right)$.

// break ties arbitrarily.

play arm $j := x(t)$, and observe reward $r_{x(t)}(t)$ at time t.

$n_j = n_j + 1$

$R_j = R_j + r_j(t)$

$t = t + 1$

Figure 2.4: UCB1 algorithm for stochastic multi-armed bandit.

Let us denote by $\hat{\mu}_i(t)$ the observed mean for arm i by time t, that is $R_i(t)/n_i(t)$, and by $\rho_i(t)$ the confidence interval $\sqrt{8 \log(t)/n_i(t)}$.

We first state some concentration bounds on the values of $\hat{\mu}_i(t)$. Using Azuma-Hoeffding inequality (Azuma, 1967; Hoeffding, 1963), we have the following (the

quality follows just from the definitions of $\hat{\mu}_i(t)$ and $\rho_i(t)$)

$$\begin{aligned}
&\mathbb{P}\big[\hat{\mu}_i(t) \notin [\mu_i - \rho_i(t), \mu_i + \rho_i(t)]\big] \\
&= \mathbb{P}\big[\hat{\mu}_i(t) \notin [\mu_i - \rho_i(t), \mu_i + \rho_i(t)]\big] \\
&\leq 2t^{-4}. \qquad (2.3.1)
\end{aligned}$$

Let us consider a sequence of T trials, and call a run of T rounds *clean* if event in (2.3.1) does not happen for any arm and any round. That is, a run of T trials is called clean if $\hat{\mu}_i(t) \in [\mu_i - \rho_i(t), \mu_i + \rho_i(t)]$ for all $i \in [K]$ and for all $t \in [T]$. A run is clean with high probability, as can be seen from the following calculation:

$$\begin{aligned}
&\mathbb{P}[\text{A run is not clean}] \\
&\leq \sum_{i=1}^{K} \sum_{t=1}^{T} \mathbb{P}[\text{Event in equation 2.3.1 happens for arm } i \text{ at time } t] \\
&\leq \sum_{i=1}^{K} \sum_{t=1}^{T} 2t^{-4} \\
&\leq \mathcal{O}(KT^{-3}).
\end{aligned}$$

In the case when the run is not clean, the algorithm can suffer a regret of as much as T, so the total regret due non-clean runs can be at most $\mathcal{O}(KT^{-2})$ in expectation. We will now focus our attention on clean runs.

Let us order the arms in the decreasing order of their average rewards μ_i. So, $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. Let us define $\Delta_i := \mu_1 - \mu_i$. The expected regret of the algorithm can be bounded by $\sum_{i=2}^{K} \mathbb{E}[n_i(T)]\Delta_i$. We will focus our attention on one particular term of this sum.

Let us bound $\mathbb{E}[n_i(T)]$ for a fixed i. Define $Q_i(T) = 32\log(T)/\Delta_i^2$. Note that after playing arm i for $Q_i(T)$ times, the confidence interval of arm i ($\rho_i(t)$) is at

most $\Delta_i/2$. We have

$$\begin{aligned}
\mathbb{E}[n_i(T)] &= \sum_{t=1}^{T} \mathbb{P}[n_i(T) \geq t] \\
&= \sum_{t=1}^{Q_i(T)} \mathbb{P}[n_i(T) \geq t] + \sum_{t=Q_i(T)+1}^{T} \mathbb{P}[n_i(T) \geq t] \\
&\leq Q_i(T) + \sum_{t=Q_i(T)+1}^{T} \mathbb{P}[n_i(T) \geq t] \\
&= Q_i(T) + \sum_{t=Q_i(T)+1}^{T} \sum_{u=t}^{T} \mathbb{P}[(n_i(u-1) = t-1) \wedge (x(u) = i)] \\
&\qquad \text{(The equality holds because the event } [n_i(T) \geq t] \\
&\qquad \text{is equal to the event that the arm was played for} \\
&\qquad t\text{-th time in some round (called round } u \text{ above).)} \\
&\leq Q_i(T) + \sum_{t=Q_i(T)+1}^{T} \sum_{u=t}^{T} \mathbb{P}\Big[(n_i(u-1) = t-1) \\
&\qquad \wedge (\hat{\mu}_i(u-1) + \rho_i(u-1) \geq \hat{\mu}_1(u-1) + \rho_1(u-1))\Big] \\
&\leq Q_i(T) + \sum_{t=Q_i(T)+1}^{T} \sum_{u=t}^{T} \mathbb{P}\Big[(n_i(u-1) = t-1) \\
&\qquad \wedge \Big(\hat{\mu}_i(u-1) + \sqrt{8(\log(u-1))/n_i(u-1)} \geq \mu_1\Big)\Big] \\
&\leq Q_i(T) + \sum_{t=Q_i(T)+1}^{T} \sum_{u=t}^{T} \mathbb{P}\Big[(n_i(u-1) = t-1) \\
&\qquad \wedge \Big(\hat{\mu}_i(u-1) + \sqrt{8(\log(T))/(t-1)} \geq \mu_1\Big)\Big] \\
&\leq Q_i(T) + \sum_{t=Q_i(T)+1}^{T} \sum_{u=t}^{T} \mathbb{P}\Big[\mu_i + \Delta_i/2 + \sqrt{8(\log(T))/Q_i(T)} \geq \mu_1\Big] \\
&\quad \text{(Since } \hat{\mu}_i(u-1) \leq \mu_i + \rho_i(u-1) \text{ for clean runs.)} \\
&\leq Q_i(T) + \sum_{t=Q_i(T)+1}^{T} \sum_{u=t}^{T} \mathbb{P}\Big[\mu_i + \Delta_i/2 + \Delta_i/2 \geq \mu_1\Big] \\
&= Q_i(T). \quad \text{(Since all the probabilities} \\
&\qquad \text{are zero.)}
\end{aligned}$$

Therefore, the total regret of the algorithm can be bounded by the sum of regret from non-clean runs, and regret from playing suboptimal arms in the clean runs. This is at most:

$$\begin{aligned}
&\mathcal{O}(KT^{-2}) + \sum_{i=2}^{K} \Delta_i \cdot \mathbb{E}[n_i(T)] \\
&\leq \mathcal{O}(KT^{-2}) + \sum_{i=2}^{K} \Delta_i \cdot Q_i(T) \\
&= \mathcal{O}(KT^{-2}) + \sum_{i=2}^{K} \frac{32 \log T}{\Delta_i}.
\end{aligned}$$

This proves the performance guarantee of the UCB1 algorithm. □

2.4 The Exp3 and Exp4 algorithms

In this section, we present algorithms for the non-stochastic version of multi-armed bandit problem (Auer et al., 2002a).

In the non-stochastic version of the problem, the rewards are not sampled using a distribution, but can be chosen arbitrarily by an adversary. The feedback model is the partial-feedback model. In summary, we have

- Arms: K arms numbered 1 through K.
- Rewards: Arbitrary rewards, bounded in $[0, 1]$.
- Feedback: partial-feedback model.

The Exp3 algorithm (presented in Figure 2.5) achieves sublinear regret for the non-stochastic multi-armed bandit problem (adaptive adversary).

We now state and prove the performance guarantee of Exp3.

PARAMETER:

$\gamma \in (0, 1)$.

VARIABLE:

$w_i(t)$ for $i \in [K]$ and $t = 1, 2, \ldots$

Set $w_i(1) = 1$ for all i.

$t = 1$.

FOR $t = 1, 2, 3, \ldots$

$w(t) = \sum_{i=1}^{K} w_i(t)$.

$p_i(t) = (1 - \gamma)\frac{w_i(t)}{w(t)} + \frac{\gamma}{K}$.

Choose the action i with probability $p_i(t)$, and set it equal to $i(t)$, and observe its reward $r_{i(t)}(t)$.

$$\hat{r}_i(t) = \begin{cases} r_i(t)/p_i(t) & \text{if } i = i(t) \\ 0 & \text{otherwise.} \end{cases}$$

$w_i(t + 1) = w_i(t) \cdot e^{\frac{\gamma}{K}\hat{r}_i(t)}$.

$t = t + 1$.

Figure 2.5: Exp3 algorithm for non-stochastic multi-armed bandit problem.

Theorem 2.4.1. *Against any adaptive adversary, the regret of* Exp3 *can be bounded by*

$$\mathbb{E}[r_{\max}(1:T)] - \mathbb{E}[r_{\mathsf{Exp3}}(1:T)] \leq \mathcal{O}(\sqrt{TK \ln K}).$$

En route to proving the above theorem, we will prove the following performance guarantee, from which the theorem follows immediately by taking $\gamma = \sqrt{K \ln K/T}$.

Lemma 2.4.2. *For any adaptive adversary, the regret of* Exp3 *can be bounded by*

$$\mathbb{E}[r_{\max}(1:T)] - \mathbb{E}[r_{\mathsf{Exp3}}(1:T)] \leq (e - 1)\gamma\mathbb{E}[r_{\max(1:T)}] + \frac{K \ln K}{\gamma}.$$

Proof. Let $F(t)$ be the σ-field generated by random choices of algorithm up to time t, and $G(t)$ be the σ-field generated by random choices of adversary up to time t. Also, let $FG(t)$ be the σ-field generated by $F(t)$ and $G(t)$.

We first analyze the evolution of weights $w(t+1)$.

$$\begin{aligned}
&\frac{\mathbb{V}[w(t+1) \mid FG(t)]}{w(t)} \\
&= \frac{1}{w(t)} \sum_{i=1}^{K} \mathbb{V}[w_i(t+1) \mid FG(t)] \\
&= \sum_{i=1}^{K} \frac{w_i(t)}{w(t)} e^{\hat{r}_i(t)\gamma/K} \\
&= \sum_{i=1}^{K} \frac{p_i(t) - \gamma/K}{1-\gamma} e^{\hat{r}_i(t)\gamma/K} \\
&\leq \sum_{i=1}^{K} \frac{p_i(t) - \gamma/K}{1-\gamma} \left(1 + \frac{\hat{r}_i(t)\gamma}{K} + \left(\frac{\hat{r}_i(t)\gamma}{K}\right)^2 (e-2)\right) \\
&\leq 1 + \frac{\gamma/K}{1-\gamma} \sum_{i=1}^{K} p_i(t)\hat{r}_i(t) + \frac{(e-2)(\gamma/K)^2}{1-\gamma} \sum_{i=1}^{K} p_i(t)\hat{r}_i(t)^2 \\
&= 1 + \frac{\gamma/K}{1-\gamma} r_{i(t)}(t) + \frac{(e-2)(\gamma/K)^2}{1-\gamma} \sum_{i=1}^{K} \hat{r}_i(t) \quad \text{(because } p_i(t)\hat{r}_i(t) \text{ is at most } r_i(t).\text{)}
\end{aligned}$$

Taking logarithms on both sides and using the inequality $1 + x \leq e^x$, we get

$$\mathbb{V}[\ln(w(t+1)) - \ln(w(t)) \mid FG(t)] \leq \frac{\gamma/K}{1-\gamma} r_{i(t)}(t) + \frac{(e-2)(\gamma/K)^2}{1-\gamma} \sum_{i=1}^{K} \hat{r}_i(t).$$

Taking the expectation with respect to randomness in round t and using linearity of expectations to get

$$\begin{aligned}
&\mathbb{E}[\ln(w(t+1)) - \ln(w(t)) \mid FG(t)] \\
&\leq \frac{\gamma/K}{1-\gamma} \mathbb{E}[r_{i(t)}(t) \mid FG(t)] + \frac{(e-2)(\gamma/K)^2}{1-\gamma} \sum_{i=1}^{K} \mathbb{E}[\hat{r}_i(t) \mid FG(t)]
\end{aligned}$$

$$= \frac{\gamma/K}{1-\gamma}\mathbb{E}[r_{i(t)}(t) \mid FG(t)] + \frac{(e-2)(\gamma/K)^2}{1-\gamma}\sum_{i=1}^{K}\mathbb{E}[r_i(t) \mid FG(t)] \quad \text{(because } \mathbb{E}[\hat{r}_i(t)] = \mathbb{E}[r_i(t)]\text{.)}$$

Taking another expectation over $FG(t)$ and summing over all t gives

$$\begin{aligned}
&\mathbb{E}[\ln(w(T+1)) - \ln(w(1))] \\
&\leq \frac{\gamma/K}{1-\gamma}\sum_{t=1}^{T}\mathbb{E}[r_{i(t)}(t)] + \frac{(e-2)(\gamma/K)^2}{1-\gamma}\sum_{t=1}^{T}\sum_{i=1}^{K}\mathbb{E}[r_i(t)] \\
&\leq \frac{\gamma/K}{1-\gamma}\mathbb{E}[r_{\mathsf{Exp3}}(1:T)] + \frac{(e-2)(\gamma/K)^2}{1-\gamma}K\mathbb{E}[r_{\max}(1:T)]. \qquad (2.4.1)
\end{aligned}$$

Also note that

$$\begin{aligned}
\mathbb{E}[\ln w(T+1)] &\geq \mathbb{E}[\ln(w_{i_0}(T+1))] \quad \text{(for every } i_0\text{.)} \\
&= \mathbb{E}[\mathbb{E}[\ln(w_{i_0}(T+1)) \mid FG(T+1)]] \\
&= \mathbb{E}\left[\frac{\gamma}{K}\sum_{t=1}^{T}\hat{r}_{i_0}(t)\right] \quad \text{(Conditioned on } FG(T+1), \ln(w_{i_0}(T+1)) \text{ is a constant.)} \\
&= \mathbb{E}\left[\frac{\gamma}{K}\sum_{t=1}^{T}r_{i_0}(t)\right] \quad \text{(because } \mathbb{E}[\hat{r}_i(t)] = \mathbb{E}[r_i(t)]\text{.)}
\end{aligned}$$

In particular, putting i_0 equal to the best action, we get

$$\mathbb{E}[\ln w(T+1)] \geq \mathbb{E}\left[\frac{\gamma}{K}\sum_{t=1}^{T}r_{\max}(t)\right] = \frac{\gamma}{K}\mathbb{E}[r_{\max}(1:T)].$$

Using these relations in (2.4.1), we get

$$\frac{\gamma}{K}\mathbb{E}[r_{\max}(1:T)] - \ln K \leq \frac{\gamma/K}{1-\gamma}\mathbb{E}[r_{\mathsf{Exp3}}(1:T)] + \frac{(e-2)(\gamma/K)^2}{1-\gamma}K\mathbb{E}[r_{\max}(1:T)].$$

Rearranging the terms, we get

$$\mathbb{E}[r_{\max}(1:T)] - \mathbb{E}[r_{\mathsf{Exp3}}(1:T)] \leq \gamma(e-1)\mathbb{E}[r_{\max}(1:T)] + (1-\gamma)\frac{K\ln K}{\gamma}.$$

We get the statement of the lemma by noting that $\gamma > 0$. □

2.4.1 Regret against best strategy from a pool

We now consider a slightly different problem, in which there are N "experts" which give "advice" to the algorithm about what to play in each round (the multi-armed bandit problem corresponds to K experts each of which "recommend" playing the corresponding arm in each time round). The goal of the algorithm is to minimize the regret with respect to the best expert in hindsight. The ingredients of the problem are:

- N experts (options) each of which chooses one of K arms in each rounds. N is typically much larger than K.
- Rewards: $[0, 1]$-bounded rewards for each arm in each time round.
- Feedback: partial-feedback model.

Let us formalize the problem now. There are N experts, each of which (indexed by superscript i) gives a probability distribution $\xi^i(t) = (\xi^i_1(t), \ldots, \xi^i_K(t))$ over the arms $j \in \{1, 2, \ldots, K\}$. The regret is defined as

$$\max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{K} \xi^i_j(t) r_j(t)\right] - \mathbb{E}[r_{\texttt{alg}}(1:T)],$$

and the goal is to minimize it.

Assumption: The *uniform-expert* (which has $\xi_j(t) = 1/K$ for all t and j) is included in the set of experts.

In Figure 2.6, we present the algorithm Exp4 for this problem, which works like Exp3. Note that throughout this section, (N) experts are indexed by superscripts, and (K) arms are indexed by subscripts.

PARAMETER:

$\gamma \in (0, 1)$.

VARIABLE:

$w^i(t)$ for $i \in [N]$ and $t = 1, 2, \ldots$

Set $w^i(1) = 1$ for all i.

$t = 1$.

FOR $t = 1, 2, 3, \ldots$

$w(t) = \sum_{i=1}^{K} w^i(t)$.

Get the expert advices $\xi^1(t), \ldots, \xi^N(t)$.

$p_j(t) = (1 - \gamma)\frac{\sum_{i=1}^{N} w^i(t)\xi_j^i(t)}{w(t)} + \frac{\gamma}{K}$.

Choose the action j with probability $p_j(t)$, and set it equal to $j(t)$, and observe its reward $r_{j(t)}(t)$.

$$\hat{r}_j(t) = \begin{cases} r_j(t)/p_j(t) & \text{if } j = j(t) \\ 0 & \text{otherwise.} \end{cases}$$

$\hat{y}^i(t) = \sum_{j=1}^{K} \xi_j^i(t)\hat{r}_j(t)$.

$w^i(t+1) = w^i(t) \cdot e^{\frac{\gamma}{K}\hat{y}^i(t)}$.

$t = t + 1$.

Figure 2.6: Exp4 algorithm for non-stochastic multi-armed bandit problem.

Lemma 2.4.3. *Against an adaptive adversary and a set of N experts, the regret of* Exp4 *can be bound by*

$$\mathbb{E}[r_{\max}(1:T)] - \mathbb{E}[r_{\text{Exp4}}(1:T)] \leq (e-1)\gamma\mathbb{E}[r_{\max}(1:T)] + \frac{K \ln N}{\gamma},$$

where the max *is taken over all N experts.*

Proof. We analyze the evolution of $w^i(t+1)$. Let us define $q^i(t) := \frac{w^i(t)}{w(t)}$.

$$\begin{aligned}
&\frac{\mathbb{V}[w(t+1) \mid FG(t)]}{w(t)} \\
&= \sum_{i=1}^{N} \frac{w^i(t)}{w(t)} e^{\hat{y}^i(t)\gamma/K} \\
&\leq \sum_{i=1}^{N} q^i(t) \left[1 + \frac{\gamma}{K}\hat{y}^i(t) + \left(\frac{\gamma}{K}\hat{y}^i(t)\right)^2 (e-2)\right] \\
&= 1 + \frac{\gamma}{K}\sum_{i=1}^{N} q^i(t)\hat{y}^i(t) + (e-2)\left(\frac{\gamma}{K}\right)^2 \sum_{i=1}^{N} q^i(t)(\hat{y}^i(t))^2.
\end{aligned}$$

Taking logarithms on both side (and making use of the inequality $1 + x \leq e^x$), we get

$$\begin{aligned}
&\mathbb{V}[\ln w(t+1) \mid FG(t)] - \ln w(t) \\
&\leq \frac{\gamma}{K}\sum_{i=1}^{N} q^i(t)\hat{y}^i(t) + (e-2)\left(\frac{\gamma}{K}\right)^2 \sum_{i=1}^{N} q^i(t)(\hat{y}^i(t))^2. \qquad (2.4.2)
\end{aligned}$$

Let us analyze each term of (2.4.2) separately. The first term of (2.4.2) can be upper bounded by

$$\begin{aligned}
\sum_{i=1}^{N} q^i(t)\hat{y}^i(t) &= \sum_{i=1}^{N} q^i(t) \sum_{j=1}^{K} \xi_j^i(t)\hat{r}_j(t) \\
&= \sum_{j=1}^{K} \hat{r}_j(t) \sum_{i=1}^{N} q^i(t)\xi_j^i(t) \\
&= \sum_{j=1}^{K} \hat{r}_j(t) \left(\frac{p_j(t) - \gamma/K}{1-\gamma}\right) \\
&\leq \sum_{j=1}^{K} \hat{r}_j(t) \left(\frac{p_j(t)}{1-\gamma}\right) \\
&= \frac{r_{j(t)}}{1-\gamma},
\end{aligned}$$

where $j(t)$ denotes the arm chosen in round t.

Similarly, the second term of (2.4.2) can be bound by

$$\sum_{i=1}^{N} q^i(t)(\hat{y}^i(t))^2 = \sum_{i=1}^{N} q^i(t) \left(\xi_{j(t)}^i(t) \cdot \hat{r}_{j(t)}(t)\right)^2$$

$$
\begin{aligned}
&\leq \hat{r}_{j(t)}(t)^2 \sum_{i=1}^{N} q^i(t)\xi^i_{j(t)}(t)^2 \\
&\leq \hat{r}_{j(t)}(t)^2 \sum_{i=1}^{N} q^i(t)\xi^i_{j(t)}(t) \\
&= \hat{r}_{j(t)}(t)^2 \frac{p_{j(t)} - \gamma/K}{1-\gamma} \\
&\leq \hat{r}_{j(t)}(t)^2 \frac{p_{j(t)}}{1-\gamma} \\
&\leq \frac{\hat{r}_{j(t)}(t)}{1-\gamma}. \quad \text{(because } p_{j(t)}(t)\hat{r}_{j(t)}(t) = \\
&\qquad\qquad\qquad\qquad r_{j(t)}(t) \leq 1.)
\end{aligned}
$$

So, (2.4.2) simplifies to

$$
\begin{aligned}
&\mathbb{V}[\ln w(t+1) \mid FG(t)] - w(t) \\
&\leq \frac{\gamma}{K}\frac{r_{j(t)}}{1-\gamma} + (e-2)\left(\frac{\gamma}{K}\right)^2 \frac{\hat{r}_{j(t)}(t)}{1-\gamma}.
\end{aligned}
$$

Take expectations (twice) on both side and sum over $t = 1, 2, \ldots, T$ to get

$$
\begin{aligned}
&\mathbb{E}[\ln w(T+1)] - w(1) \\
&\leq \frac{\gamma}{K(1-\gamma)} \sum_{t=1}^{T} \mathbb{E}[r_{j(t)}] + \frac{(e-2)}{1-\gamma}\left(\frac{\gamma}{K}\right)^2 \mathbb{E}[\hat{r}_{j(t)}(t)] \\
&= \frac{\gamma}{K(1-\gamma)} \sum_{t=1}^{T} \mathbb{E}[r_{j(t)}(t)] + \frac{(e-2)}{1-\gamma}\left(\frac{\gamma}{K}\right)^2 \sum_{t=1}^{T} \mathbb{E}[r_{j(t)}(t)]. \qquad (2.4.3)
\end{aligned}
$$

To simplify the left hand side of (2.4.3), note that for any $i_0 \in [N]$, we have (where $y^i(t)$ is defined as the reward of expert i in round t)

$$
\begin{aligned}
\mathbb{E}[\ln w(T+1)] &\geq \mathbb{E}[\ln w^{i_0}(T+1)] \\
&= \mathbb{E}\left[\frac{\gamma}{K}\sum_{t=1}^{T} \hat{y}^{i_0}(t)\right] \\
&= \mathbb{E}\left[\frac{\gamma}{K}\sum_{t=1}^{T}\sum_{j=1}^{K} \xi_j^{i_0}(t)\cdot\hat{r}_j(t)\right] \\
&= \mathbb{E}\left[\frac{\gamma}{K}\sum_{t=1}^{T}\sum_{j=1}^{K} \xi_j^{i_0}(t)\cdot r_j(t)\right]
\end{aligned}
$$

$$= \mathbb{E}\left[\frac{\gamma}{K}\sum_{t=1}^{T} y^{i_0}(t)\right].$$

The first term on the right hand side of (2.4.3) can be bounded by

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{E}[r_{j(t)}(t)] &= K\cdot\frac{1}{K}\sum_{t=1}^{T}\mathbb{E}[r_{j(t)}(t)] \\
&\leq K\cdot\frac{1}{K}\sum_{t=1}^{T}\sum_{j=1}^{K}\mathbb{E}[r_j(t)] \\
&= K\cdot\mathbb{E}[y^{\text{uniform expert}}(1:T)] \\
&\leq K\cdot\mathbb{E}[y^{\max}(1:T)]. \quad \text{(because, the uniform expert is in the set of experts.)}
\end{aligned}$$

Equation 2.4.3 hence simplifies to

$$\frac{\gamma}{K}\mathbb{E}[y^{\max}(1:T)] - \ln N \leq \frac{\gamma/K}{1-\gamma}\mathbb{E}[r_{\mathsf{Exp4}}(1:T)] + \frac{(e-2)}{1-\gamma}\left(\frac{\gamma}{K}\right)^2 K\cdot\mathbb{E}[y^{\max(1:T)}].$$

Rearranging the terms, we get

$$\mathbb{E}[y^{\max}(1:T)] - \mathbb{E}[r^{\mathsf{Exp4}}(1:T)] \leq (e-1)\gamma\mathbb{E}[y^{\max}(1:T)] + (1-\gamma)\frac{K\ln N}{\gamma}.$$

This proves the statement of the lemma. □

It is worth noting here that the performance guarantee of Exp4 algorithm also holds against adaptive adversaries.

CHAPTER 3

TRUTHFUL MULTI-ARMED BANDIT PROBLEM

In this chapter, we consider a multi-round auction setting motivated by pay-per-click auctions for Internet advertising. In each round the auctioneer selects an advertiser and shows her ad, which is then either clicked or not. An advertiser derives value from clicks; the value of a click is her private information. Initially, neither the auctioneer nor the advertisers have any information about the likelihood of clicks on the advertisements. The auctioneer's goal is to design a (dominant strategy) truthful mechanism (one in which each advertiser prefer to tell the truth about her private information, see Section 3.4) that (approximately) maximizes the social welfare.

If the advertisers bid their true private values, our problem is equivalent to the multi-armed bandit problem, and thus can be viewed as a strategic version of the latter. In particular, for both problems the quality of an algorithm can be characterized by regret, the difference in social welfare between the algorithm and the benchmark which always selects the same "best" advertisement. We investigate how the design of multi-armed bandit algorithms is affected by the restriction that the resulting mechanism must be truthful. We find that truthful mechanisms have certain strong structural properties – essentially, they must separate exploration from exploitation – *and* they incur much higher regret than the optimal multi-armed bandit algorithms. Moreover, we provide a truthful mechanism

which (essentially) matches our lower bound on regret.

3.1 Introduction

In recent years there has been much interest in understanding the implication of strategic behavior on the performance of algorithms whose input is distributed among selfish agents. This study was mainly motivated by the Internet, the main arena of large scale interaction of agents with conflicting goals. The field of Algorithmic Mechanism Design (Nisan and Ronen, 2001) studies the design of mechanisms in computational settings (for background see the recent book Nisan et al. (2007) and survey Roughgarden (2008)).

Much attention has been drawn to the market for sponsored search (e.g. Lahaie et al. (2007); Edelman et al. (2007); Varian (2007); Mehta et al. (2007); Aggarwal and Muthukrishnan (2008)), a billion dollar market with numerous auctions running every second. Research on sponsored search mostly focus on equilibria of the Generalized Second Price (GSP) auction (Edelman et al., 2007; Varian, 2007), the auction that is most commonly used in practice (e.g. by Google and Yahoo), or on the design of truthful auctions (Aggarwal et al., 2006). All these auctions rely on knowing the rates at which users click on the different advertisements (a.k.a. Click-Through-Rates, or CTRs), and do not consider the process in which these CTRs are learned or refined over time by observing users' behavior. We argue that strategic agents would take this process into account, as it influences their utility. Prior work (Immorlica et al., 2005) focused on the implication of click fraud on the methods used to learn CTRs. We, on the other hand, are interested in the implications of the *strategic bidding* by the agents. Thus, we consider the problem of designing truthful sponsored search auctions when the process of learning the

CTRs is a part of the game.

We are mainly interested in the interplay between the online learning and the strategic aspects of the problem. To isolate this issue, we consider the following setting, which is a natural *strategic* version of the multi-armed bandit (MAB) problem introduced in Chapter 1. In this setting, there are k agents (we use small k to denote the number of agents in this chapter, as opposed to capital K which was used earlier in Chapter 1). Each agent i has a single advertisement, and a *private* value $v_i > 0$ for every click she gets. The mechanism is an online algorithm that first solicits bids from the agents, and then runs for T rounds. In each round the mechanism picks an agent (using the bids and the clicks observed in the past rounds), displays her advertisement, and receives a feedback – if there was a click or not. Payments are assigned after round T. Each agent tries to maximize her own utility: the difference between the value that she derives from clicks and the payment she pays. We assume that initially no information is known about the likelihood of each agent to be clicked, and in particular there are no Bayesian priors.

We are interested in designing mechanisms which are truthful (in *dominant strategies*): every agent maximizes her utility by bidding truthfully, for any bids of the others and *for any clicks* that would have been received. The goal is to maximize the social welfare.[1] Since the payments cancel out, this is equivalent to maximizing the total value derived from clicks, where an agent's contribution to that total is her private value times the number of clicks she receives. We call this setting the *MAB mechanism design problem*.

[1]Social welfare includes both the auctioneer's revenue and the agents' utility. Since in practice different sponsored search platforms compete against one another, taking into account the agents' utility increases the platform's attractiveness to the advertisers.

In the absence of strategic behavior this problem reduces to a standard MAB formulation in which an algorithm repeatedly chooses one of the k alternatives ("arms") and observes the associated payoff: the value-per-click of the corresponding ad if the ad is clicked, and 0 otherwise. The crucial aspect in MAB problems is the tradeoff between acquiring more information (*exploration*) and using the current information to choose a good agent (*exploitation*). MAB problems have been studied intensively for the past three decades (see (Berry and Fristedt, 1985; Cesa-Bianchi and Lugosi, 2006; Gittins, 1989)). In particular, the above formulation is well-understood (Auer et al., 2002b,b; Dani and Hayes, 2006) in terms of regret relative to the benchmark which always chooses the same "best" alternative. This notion of regret naturally extends to the strategic setting outlined above, the total payoff being exactly equal to the social welfare, and the regret being exactly the loss in social welfare. Thus one can directly compare MAB algorithms and MAB mechanisms in terms of welfare loss (regret).

Broadly, we ask how the design of MAB algorithms is affected by the restriction of truthfulness: what is the difference between the best *algorithms* and the best *truthful mechanisms*? We are interested both in terms of the structural properties and the gap in performance (in terms of regret). We are not aware of any prior work that characterizes truthful learning algorithms or proves negative results on their performance.

3.2 Our contributions

We present two main contributions.

- First, we present a characterization of (dominant-strategy) truthful mecha-

nisms.

- Second, we present a lower bound on the regret that such mechanisms must suffer. This regret is significantly larger than the regret of the best MAB algorithms.

Formally, a mechanism for the MAB mechanism design problem is a pair $(\mathcal{A}, \mathcal{P})$, where $\mathcal{A}$ is the *allocation rule* (essentially, an MAB algorithm), and $\mathcal{P}$ is the *payment rule*. Note that regret is completely determined by the allocation rule. As is standard in the literature, we focus on mechanisms in which each agent's payment (averaged over clicks) is between 0 and her bid; such mechanisms are called *normalized*, and they satisfy voluntary participation (agents don't have any incentives not to participate).

The setting we study is a *single-parameter auction*, the most studied and well-understood type of auctions. For such settings truthful mechanisms are fully characterized (Myerson, 1981; Archer and Tardos, 2001): a mechanism is truthful if and only if the allocation rule is monotone (by increasing her bid an agent cannot cause a decrease in the number of clicks she gets), and the payment rule is defined in a specific and (essentially) unique way. Yet, this characterization is *not* the right characterization for the MAB setting! The main problem is that in our setting click information for any agent that is not chosen at a given round is not available to the mechanism, and thus cannot be used in the computation of payments. Thus, the payment cannot depend on any unobserved clicks. We show that this has severe implications on the structure of truthful mechanisms.

The first notable property of a truthful mechanism is a much stronger version of monotonicity:

Definition 3.2.1. A *realization* consists of click information for all agents at all

rounds (including unobserved ones). An allocation rule is *pointwise monotone* if for each realization, each bid profile and each round, if an agent is played/shown at the round, then she is also played/shown after increasing her bid (fixing everything else).

Let us consider (for the ease of exposition) allocation rules that satisfy the following two natural conditions. First, an allocation rule is *scale-free* if it is invariant under multiplying all bids by the same positive number (essentially, changing the currency unit). Second, it is *independent of irrelevant alternatives* (*IIA*, for short) if for any given realization, bid profile and round, a change of bid of agent i cannot transfer the allocation in this round from agent j to agent l, where these are three distinct agents.

We show that any truthful mechanism must have a strict separation between exploration and exploitation. A crucial feature of exploration is the ability to influence the allocation in forthcoming rounds. To make this point more concrete, we call a round *influential* for a given realization if for some bid profile changing the realization for this round can affect the allocation in some future round. We show that in any such round, the allocation can not depend on the bids. Thus, influential rounds are essentially useless for exploitation.

Definition 3.2.2. An allocation rule $\mathcal{A}$ is called *exploration-separated* if for any given realization, the allocation in any influential round for that realization does not depend on the bids.

We are now ready to present our main structural result, which is in fact a complete characterization.

Theorem 3.2.3. *Consider the MAB mechanism design problem. Let $\mathcal{A}$ be a non-*

degenerate[2] *deterministic allocation rule which is scale-free and satisfies IIA. Then mechanism* $(\mathcal{A}, \mathcal{P})$ *is normalized and truthful for some payment rule* $\mathcal{P}$ *if and only if* $\mathcal{A}$ *is pointwise monotone and exploration-separated.*

We also obtain a similar (but somewhat more complicated) characterization without assuming that allocations are scale-free and satisfy IIA (Theorem 3.5.8). We use it then to derive Theorem 3.2.3. We emphasize that our characterization results hold regardless of whether the auctioneer's goal is to maximize welfare or revenue or any other objective.

In view of Theorem 3.2.3, we present a lower bound on the performance of exploration-separated algorithms. We consider a setting, termed the *stochastic MAB mechanism design problem*, in which each click on a given advertisement is an independent random event which happens with a fixed probability, a.k.a. the CTR. The expected "payoff" from choosing a given agent is her private value times her CTR. For the ease of exposition, assume that the bids lie in the interval $[0, 1]$. Then the non-strategic version is the *stochastic MAB problem* in which the payoff from choosing a given arm i is an independent sample in $[0, 1]$ with a fixed mean μ_i. In both versions, *regret* is defined with respect to a hypothetical allocation rule (resp. algorithm) that always chooses an arm with the maximal expected payoff. Specifically, regret is the expected difference between the social welfare (resp. total payoff) of the benchmark and that of the allocation rule (resp. algorithm). The goal is to minimize $R(T)$, worst-case regret over all problem instances on T rounds.

[2]Non-degeneracy is a mild technical assumption, formally defined in Section 3.4, which ensures that (essentially) if a given allocation happens for some bid profile (b_i, b_{-i}) then the same allocation happens for all bid profiles (x, b_{-i}), where x ranges over some non-degenerate interval. Without this assumption, all structural results hold (essentially) *almost surely* w.r.t the k-dimensional Lebesgue measure on the bid vectors. Exposition becomes significantly more cumbersome, yet leads to the same lower bounds on regret. For clarity, we assume non-degeneracy throughout this version of the paper.

We show that the worst-case regret of any exploration-separated mechanism is *larger* than that of the optimal MAB algorithm (Auer et al., 2002b): $\Omega(T^{2/3})$ vs $O(\sqrt{T})$ for a fixed number of agents. We obtain an even more pronounced difference if we restrict our attention to the *δ-gap* problem instances: instances for which the best agent is better than the second-best by a (comparatively large) amount δ, that is $\mu_1 v_1 - \mu_2 v_2 = \delta \cdot (\max_i v_i)$, where arms are arranged such that $\mu_1 v_1 \geq \mu_2 v_2 \geq \cdots \geq \mu_k v_k$. Such instances are known to be easy for the MAB algorithms. Namely, an algorithm can achieve the optimal worst-case regret $O(\sqrt{kT \log T})$ *and* regret $O(\frac{k}{\delta} \log T)$ on δ-gap instances (Lai and Robbins, 1985a; Auer et al., 2002b). However, for exploration-separated mechanisms the worst-case regret $R_\delta(T)$ over the δ-gap instances is polynomial in T as long as worst-case regret is even remotely non-trivial (i.e., sublinear). Thus, for the δ-gap instances the gap between algorithms and truthful mechanisms in the worst-case regret is *exponential* in T.

Theorem 3.2.4. *Consider the stochastic MAB mechanism design problem with k agents. Let $\mathcal{A}$ be a deterministic allocation rule that is exploration-separated. Then $\mathcal{A}$ has worst-case regret $R(T) = \Omega(k^{1/3} T^{2/3})$. Moreover, if $R(T) = O(T^\gamma)$ for some $\gamma < 1$ then for every fixed $\delta \leq \frac{1}{4}$ and $\lambda < 2(1-\gamma)$ the worst-case regret over the δ-gap instances is $R_\delta(T) = \Omega(\delta\, T^\lambda)$.*

We note that our lower bounds holds for a more general setting in which the values-per-click can change over time, and the advertisers are allowed to change their bids at every time step.

To complete the picture, we present a very simple (deterministic) mechanism that is truthful and normalized, and matches the lower bound $R(T) = \Omega(k^{1/3} T^{2/3})$ up to logarithmic factors.

We also provide a number of extensions. First, we prove a similar (but slightly weaker) regret bound without the scale-free assumption. Second, we extend some of our results to randomized mechanisms; in this setting, (dominant-strategy) truthfulness means "truthfulness for each realization of the private randomness". Third, we consider a weaker notion of truthfulness for randomized mechanisms – for each realization of the clicks, but in expectation over the random seed, and use this notion to provide algorithmic results for the version of the MAB mechanism design problem in which clicks are chosen by an adversary. Fourth, we discuss an even more permissive notion of truthfulness – truthfulness in expectation over the clicks (and the random seed).

3.3 Other related work and discussion

The question of how the performance of a truthful mechanism compares to that of the optimal algorithm for the corresponding non-strategic problem has been considered in the literature in a number of other auction settings. Performance gaps have been shown for various scheduling problems (Archer and Tardos, 2001; Nisan and Ronen, 2001; Dobzinski and Sundararajan, 2008) and for online auction for expiring goods (Lavi and Nisan, 2005). Other papers presented approximation gaps due to *computational constraints*, e.g. for combinatorial auctions (Lavi et al., 2003; Dobzinski and Sundararajan, 2008) and combinatorial public projects (Papadimitriou et al., 2008), showing a gap via a structural result for truthful mechanisms.

The study of MAB mechanisms has been initiated by Gonen and Pavlov (Gonen and Pavlov, 2007). The authors present a MAB mechanism which is claimed to be truthful in a certain approximate sense. Unfortunately, this mechanism does not satisfy the claimed properties; this was also confirmed with the authors through

personal communication (see also a similar note in (Devanur and Kakade, 2009)).

MAB algorithms were used in the design of Cost-Per-Action sponsored search auctions in Nazerzadeh et al. (2008), where the authors construct a mechanism with approximate properties of truthfulness and individual rationality. Approximately truthful mechanisms are reasonable assuming the agents would not lie unless it leads to significant gains. However, this solution concept is weaker than the exact notion and it may still be rational for the agents to deviate (perhaps significantly) from being truthful. Moreover, as truthful bidding is not a Nash equilibrium, agents might have an increased incentive to deviate if they speculate that others are deviating. All of that may result in unpredictable, and possibly highly suboptimal outcomes. In this work we focus on understanding what can be achieved with the *exact* truthfulness, mainly proving results of structural and lower-bounding nature. We note in passing that providing similar results for the approximately truthful setting such as the one in Nazerzadeh et al. (2008) is a worthy and challenging open question.

Independently and concurrently, Devanur and Kakade (2009) have studied truthful MAB mechanisms with focus on maximizing the revenue. They present a lower bound of $\Omega(T^{2/3})$ on the loss in revenue with respect to the VCG (Vickrey-Clarke-Groves) payment, as well as a truthful mechanism that matches the lower bound. (This mechanism is almost identical to the one that we present in order to match the lower bound in Theorem 3.6.1.)

Our lower bounds use (a novel application of) the relative entropy technique from (Lai and Robbins, 1985a; Auer et al., 2002b), see (Kleinberg, 2007b) for an account. For other application of this technique, see e.g. (Dani and Hayes, 2006; Karp and Kleinberg, 2007a; Kleinberg et al., 2008; Ben-Or and Hassidim, 2008).

Our work focuses on regret in a prior-free setting in which the algorithm has no prior on CTRs. This is in contrast to the recent line of work on *dynamic auctions* (Bergemann and Välimäki, 2006; Athey and Segal, 2007) which considers fully Bayesian settings in which there is a known prior on CTRs, and VCG-like social welfare-maximizing mechanisms are feasible. In our prior-free setting VCG-mechanisms cannot be applied as such mechanisms require the allocation to exactly maximize the expected social welfare, which is impossible (and not well-defined) without a prior.

We require the mechanisms to satisfy a strong notion of truthfulness: bidding truthful is optimal for *every* possible realization (and bids of others). This notion is attractive as it does not require the agents to be risk neutral. Moreover, it allows for the CTRs to change over time (and still incentivizes agents to be truthful). Finally, an agent never regrets in retrospect that she has been truthful. It is desirable to understand this notion before moving to weaker notions.

Map of the chapter. Section 3.4 is preliminaries. Truthfulness characterization is developed and proved in Section 3.5. The lower bounds on regret is in Section 3.6 and a simple mechanism that matches them is in Section 3.7. Various extensions are discussed in Section 3.8.

3.4 Definitions and preliminaries

In the MAB mechanism design problem, there is a set K of k agents numbered from 1 to k. Each agent i has a *value* $v_i > 0$ for every click she gets; this value is known only to agent i. Initially, each agent i submits a *bid* $b_i > 0$, possibly different

from v_i.[3] The "game" lasts for T rounds, where T is the given *time horizon*. A *realization* represents the click information for all agents and all rounds. Formally, it is a tuple $\rho = (\rho_1, \ldots, \rho_k)$ such that for every agent i and round t, the bit $\rho_i(t) \in \{0, 1\}$ indicates whether i gets a click if played at round t. An *instance* of the MAB mechanism design problem consists of the number of agents k, time horizon T, a vector of private values $v = (v_1, \ldots, v_k)$, a vector of bids (*bid profile*) $b = (b_1, \ldots, b_k)$, and realization ρ.

A *mechanism* is a pair $(\mathcal{A}, \mathcal{P})$, where $\mathcal{A}$ is allocation rule and $\mathcal{P}$ is the payment rule. An *allocation rule* is represented by a function $\mathcal{A}$ that maps bid profile b, realization ρ and a round t to the agent i that is chosen (receives an *impression*) in this round: $\mathcal{A}(b; \rho; t) = i$. We also denote $\mathcal{A}_i(b; \rho; t) = \mathbf{1}_{\{\mathcal{A}(b;\rho;t)=i\}}$. The allocation is *online* in the sense that at each round it can only depend on clicks observed prior to that round. Moreover, it does not know the realization in advance; in every round it only observes the realization for the agent that is shown in that round. A *payment rule* is a tuple $\mathcal{P} = (\mathcal{P}_1, \ldots, \mathcal{P}_k)$, where $\mathcal{P}_i(b; \rho) \in \mathbb{R}$ denotes the payment charged to agent i when the bids are b and the realization is ρ.[4] Again, the payment can only depends on observed clicks. A mechanism is called *normalized* if for any agent i, bids b and realization ρ it holds that $\mathcal{P}_i(b; \rho)$ is non-negative and at most b_i times the number of clicks agent i got.

For given realization ρ and bid profile b, the number of clicks received by agent i is denoted $\mathcal{C}_i(b; \rho)$. Call $\mathcal{C} = (\mathcal{C}_1, \ldots, \mathcal{C}_k)$ the *click-allocation* for $\mathcal{A}$. The *utility*

[3] One can also consider a more realistic and general model in which the value-per-click of an agent changes over time and the agents are allowed to change their bid at every round. The case that the value-per-click of each agent does not change over time is a special case. In that case truthfulness implies that each agent basically submits one bid as in our model (the same bid at every round), thus our main results (necessary conditions for truthfulness and regret lower bounds) also hold for the more general model.

[4] We allow the mechanism to determine the payments at the end of the T rounds, and not after every round. This makes that task of designing a truthful mechanism *easier* and thus strengthen our necessary condition for truthfulness (the condition used to derive the lower bounds on regret.)

that agent i with value v_i gets from the mechanism $(\mathcal{A}, \mathcal{P})$ when the bids are b and the realization is ρ is $\mathcal{U}_i(v_i; b; \rho) = v_i \cdot \mathcal{C}_i(b; \rho) - \mathcal{P}_i(b; \rho)$ (called *quasi-linear* utility). The mechanism is *truthful* if for any agent i, value v_i, bid profile b and realization ρ it is the case that $\mathcal{U}_i(v_i; v_i, b_{-i}; \rho) \geq \mathcal{U}_i(v_i; b_i, b_{-i}; \rho)$.

In the *stochastic* MAB mechanism design problem, an adversary specifies a vector $\mu = (\mu_1, \ \ldots \ , \mu_k)$ of CTRs (concealed from $\mathcal{A}$), then for each agent i and round t, realization $\rho_i(t)$ is chosen independently with mean μ_i. Thus, an instance of the problem includes μ rather than a fixed realization. For a given problem instance $\mathcal{I}$, let $i^* \in \operatorname{argmax}_i \mu_i v_i$, then *regret* on this instance is defined as

$$R^{\mathcal{I}}(T) = T\, v_{i^*} \mu_{i^*} - \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i=1}^{k} \mu_i\, v_i\, \mathcal{A}_i(b;\, \rho;\, t) \right]. \tag{3.4.1}$$

For a given parameter $v_{\max}$, the *worst-case regret*[5] $R(T; v_{\max})$ denotes the supremum of $R^{\mathcal{I}}(T)$ over all problem instances $\mathcal{I}$ in which all private values are at most $v_{\max}$. Similarly, we define $R_\delta(T; v_{\max})$, the *worst-case δ-regret*, by taking the supremum only on instances with δ-gap.

Most of our results are stated for *non-degenerate* allocation rules, defined as follows. An interval is called *non-degenerate* if it has positive length. Fix bid profile b, realization ρ, and rounds t and t' with $t \leq t'$. Let $i = \mathcal{A}(b; \rho; t)$ and ρ' be the allocation obtained from ρ by flipping the bit $\rho_i(t)$. An allocation rule $\mathcal{A}$ is *non-degenerate* w.r.t. (b, ρ, t, t') if there exists a non-degenerate interval I containing b_i such that

$$\mathcal{A}_i(b'_i, b_{-i}; \varphi; s) = \mathcal{A}_i(b; \varphi; s) \quad \text{for each } \varphi \in \{\rho, \rho'\}, \text{ each } s \in \{t, t'\}, \text{ and all } b'_i \in I.$$

In essence, non-degeneracy requires that there is a small enough interval containing b_i such that the allocation for agent i is same in all rounds irrespective of i's bid

[5]By abuse of notation, when clear from the context, the "worst-case regret" is sometimes simply called "regret".

as long as it is in the interval. An allocation rule is *non-degenerate* if it is non-degenerate w.r.t. each tuple (b, ρ, t, t').

3.5 Truthfulness characterization

Before presenting our characterization we begin by describing some related background. The click allocation $\mathcal{C}$ is *non-decreasing* if for each agent i, increasing her bid (and keeping everything else fixed) does not decrease $\mathcal{C}_i$. Prior work has established a characterization of truthful mechanisms for single-parameter domains (domains in which the private information of each agent is one-dimensional), relating click allocation monotonicity and truthfulness (see below). For our problem, this result is a characterization of MAB algorithms that are truthful for a given realization ρ, assuming that the *entire* realization ρ can be used to compute payments (when computing payments one can use click information for every round and every agent, even if the agent was not shown at that round.) One of our main contributions is a characterization of MAB allocation rules that can be truthfully implemented when payment computation is restricted to only use clicks information of the actual impressions assigned by the allocation rule.

An MAB allocation rule $\mathcal{A}$ is *truthful with unrestricted payment computation* if it is truthful with a payment rule that can use the *entire* realization ρ in it computation. We next present the prior result characterizing truthful mechanisms with unrestricted payment computation.

Theorem 3.5.1 (Myerson (1981), Archer and Tardos (2001))**.** *Let $(\mathcal{A}, \mathcal{P})$ be a normalized mechanism for the MAB mechanism design problem. It is truthful with unrestricted payment computation if and only if for any given realization ρ the*

corresponding click-allocation $\mathcal{C}$ is non-decreasing and the payment rule is given by

$$\mathcal{P}_i(b_i, b_{-i}; \rho) = b_i \cdot \mathcal{C}_i(b_i, b_{-i}; \rho) - \int_0^{b_i} \mathcal{C}_i(x, b_{-i}; \rho)\, dx. \tag{3.5.1}$$

We can now move to characterize truthful MAB mechanisms when the payment computation is restricted. The following notation will be useful: for a given realization ρ, let $\rho \oplus \mathbf{1}(i,t)$, be the realization that coincides with ρ everywhere, except that the bit $\rho_i(t)$ is flipped.

The first notable property of truthful mechanisms is a stronger version of monotonicity. Recall (see Definition 3.2.1) that an allocation rule $\mathcal{A}$ is *pointwise monotone* if for each realization ρ, bid profile b, round t and agent i, if $\mathcal{A}_i(b_i, b_{-i}; \rho; t) = 1$ then $\mathcal{A}_i(b_i^+, b_{-i}; \rho; t) = 1$ for any $b_i^+ > b_i$. In words, increasing a bid cannot cause a loss of an impression.

Lemma 3.5.2. *Consider the MAB mechanism design problem. Let $(\mathcal{A}, \mathcal{P})$ be a normalized truthful mechanism such that $\mathcal{A}$ is a non-degenerate deterministic allocation rule. Then $\mathcal{A}$ is pointwise-monotone.*

Proof. For a contradiction, assume not. Then there is a realization ρ, a bid profile b, a round t and agent i such that agent i loses an impression in round t by increasing her bid from b_i to some larger value b_i^+. In other words, we have $\mathcal{A}_i(b_i^+, b_{-i}; \rho; t) < \mathcal{A}_i(b_i, b_{-i}; \rho; t)$. Without loss of generality, let us assume that there are no clicks after round t, that is $\rho_j(t') = 0$ for any agent j and any round $t' > t$ (since changes in ρ after round t does not affect anything before round t).

Let $\rho' = \rho \oplus \mathbf{1}(i,t)$. The allocation in round t cannot depend on this bit, so it must be the same for both realizations. Now, for each realization $\varphi \in \{\rho, \rho'\}$ the mechanism must be able to compute the price for agent i when bids are (b_i^+, b_{-i}).

That involves computing the integral $I_i(\varphi) = \int_{x \le b_i^+} \mathcal{C}_i(x, b_{-i}; \varphi)\, dx$ from (3.5.1). We claim that $I_i(\rho) \neq I_i(\rho')$. However, the mechanism cannot distinguish between ρ and ρ' since they only differ in bit (i,t) and agent i does not get an impression in round t. This is a contradiction.

It remains to prove the claim. Without loss of generality, assume that $\rho_i(t) = 0$ (otherwise interchange the role of ρ and ρ'). We first note that $\mathcal{C}_i(x, b_{-i}; \rho) \le \mathcal{C}_i(x, b_{-i}; \rho')$ for every x. This is because everything is same in ρ and ρ' until round t (so the impressions are same too), there are no clicks after round t, and in round t the behavior of $\mathcal{A}$ on the two realizations can be different only if that agent i gets an impression, in which case she is clicked under ρ' and not clicked under ρ.

Since $\mathcal{A}$ is non-degenerate, there exists a non-degenerate interval I containing b_i such that changing bid of agent i to any value in this interval does not change the allocation at round t (both for ρ and for ρ'). For any $x \in I$ we have $\mathcal{C}_i(x, b_{-i}; \rho) < \mathcal{C}_i(x, b_{-i}; \rho')$, where the difference is due to the click in round t. It follows that $I_i(\rho) < I_i(\rho')$. Claim proved. Hence, the mechanism cannot be implemented truthfully. □

Recall (see Definition 3.2.2) that round t is *influential* for a given realization ρ if for some bid profile b there exists a round $t' > t$ such that $\mathcal{A}(b; \rho; t') \neq \mathcal{A}(b; \rho \oplus \mathbf{1}(j,t); t')$ for $j = \mathcal{A}(b; \rho; t)$. In words: changing the relevant part of the realization at round t affects the allocation in some future round t'. An allocation rule $\mathcal{A}$ is called *exploration-separated* if for any given realization ρ and round t that is influential for ρ, it holds that $\mathcal{A}(b; \rho; t) = \mathcal{A}(b'; \rho; t)$ for any two bid vectors b, b' (allocation at t does not depend on the bids).

The main structural implication is "truthful implies exploration-separated".

To illustrate the ideas behind this implication, we first state and prove it for two agents.

Proposition 3.5.3. *Consider the MAB mechanism design problem with two agents. Let $\mathcal{A}$ be a non-degenerate scale-free deterministic allocation rule. If $(\mathcal{A}, \mathcal{P})$ is a normalized truthful mechanism for some $\mathcal{P}$, then it is exploration separated.*

Proof. Assume $\mathcal{A}$ is not exploration-separated. Then there is a *counterexample* (ρ, t): a realization ρ and a round t such that round t is influential and allocation in round t depends on bids. We want to prove that this leads to a contradiction.

Let us pick a counterexample (ρ, t) with some useful properties. Since round t is influential, there exists a realization ρ and bid profile b such that the allocation at some round $t' > t$ (the *influenced* round) is different under realization ρ and another realization $\rho' = \rho \oplus \mathbf{1}(j, t)$, where $j = \mathcal{A}(b; \rho; t)$ is the agent chosen at round t under ρ. Without loss of generality, let us pick a counterexample with minimum value of t' over all choices of (b, ρ, t). For ease of exposition, from this point on let us assume that $j = 2$. For the counterexample we can also assume that $\rho_1(t') = 1$, and that there are no clicks after round t', that is $\rho_l(t'') = \rho'_l(t'') = 0$ for all $t'' > t'$ and for all $l \in \{1, 2\}$.

We know that the allocation in round t depends on bids. This means that agent 1 gets an impression in round t for some bid profile $\hat{b} = (\hat{b}_1, \hat{b}_2)$ under realization ρ, that is $\mathcal{A}(\hat{b}; \rho; t) = 1$. As the mechanism is scale-free this means that, denoting $b_1^+ = \hat{b}_1\, b_2/\hat{b}_2$ we have $\mathcal{A}(b_1^+, b_2; \rho; t) = 1$. Since $\mathcal{A}(b_1, b_2; \rho; t) = 2$ and $\mathcal{A}(b_1^+, b_2; \rho; t) = 1$, pointwise monotonicity (Lemma 3.5.2) implies that $b_1^+ > b_1$. We conclude that there exists a bid $b_1^+ > b_1$ for agent 1 such that $\mathcal{A}(b_1^+, b_2; \rho; t) = 1$.

Now, the mechanism needs to compute prices for agent 1 for bids (b_1^+, b_2)

under realizations ρ and ρ', that is $\mathcal{P}_1(b_1^+, b_2; \rho)$ and $\mathcal{P}_1(b_i^+, b_2; \rho')$. Therefore, the mechanism needs to compute the integral $I_1(\varphi) = \int_{x \le b_1^+} \mathcal{C}_1(x, b_2; \varphi)\, dx$ for both realizations $\varphi \in \{\rho, \rho'\}$.

First of all, for all $x \le b_1^+$ and for all $t'' < t'$, $\mathcal{A}(x, b_2; \rho; t'') = \mathcal{A}(x, b_2; \rho'; t'')$, since otherwise the minimality of t' will be violated. The only difference in the allocation can occur in round t'.

Let us assume $\mathcal{A}_1(b_1, b_2; \rho; t') < \mathcal{A}_1(b_1, b_2; \rho', t')$ (otherwise, we can swap ρ and ρ'). We make the claim that for all bids $x \le b_1^+$ of agent 1, the influence of round t on round t' is in the same "direction":

$$\mathcal{A}_1(x, b_2; \rho; t') \le \mathcal{A}_1(x, b_2; \rho'; t') \text{ for all } x \le b_1^+. \tag{3.5.2}$$

Suppose (3.5.2) does not hold. Then there is an $x < b_1^+$ such that $1 = \mathcal{A}_1(x, b_2; \rho; t') > \mathcal{A}_1(x, b_2; \rho'; t') = 0$. (Note that we have used the fact that the mechanism is deterministic.) If $x < b_1$ then pointwise monotonicity is violated under realization ρ, since $\mathcal{A}_1(x, b_2; \rho; t') > A_1(b_1, b_2; \rho; t')$; otherwise it is violated under realization ρ', giving a contradiction in both cases. The claim (3.5.2) follows.

Since $\mathcal{A}$ is non-degenerate, there exists a non-degenerate interval I containing b_i such that if agent 1 bids any value $x \in I$ then $\mathcal{A}_1(x, b_2; \rho; t') < \mathcal{A}_1(x, b_2; \rho'; t')$. Now by (3.5.2) it follows that $I_1(\rho) < I_2(\rho')$. However, the mechanism cannot distinguish between ρ and ρ' when the bid of agent 1 is b_1^+, since the differing bit $\rho_2(t)$ is not observed. Therefore the mechanism cannot compute prices, contradiction. □

3.5.1 General Truthfulness Characterization

Let us develop the general truthfulness characterization that does not assume that an allocation is scale-free and IIA. We will later use it to derive Theorem 3.2.3.

Definition 3.5.4. Fix realization ρ and bid vector b. A round t is called $(b;\rho)$-*secured* from agent i if $\mathcal{A}(b_i^+, b_{-i};\rho;t) = \mathcal{A}(b_i, b_{-i};\rho;t)$ for any $b_i^+ > b_i$. A round t is called *bid-independent* w.r.t. ρ if the allocation $\mathcal{A}(b;\rho;t)$ is a constant function of b.

The following definitions elaborate on the notion of an *influential round*.

Definition 3.5.5. A round t is called $(b;\rho)$-*influential*, for bid profile b and realization ρ, if for some round $t' > t$ it holds that $\mathcal{A}(b;\rho;t') \neq \mathcal{A}(b;\rho';t')$ for realization $\rho' = \rho \oplus \mathbf{1}(j,t)$ such that $j = \mathcal{A}(b;\rho;t)$.[6] In this case, t' is called the *influenced round* and j is called the *influencing agent* of round t. The agent i is called an *influenced agent* of round t if $i \in \{\mathcal{A}(b;\rho;t'),\ \mathcal{A}(b;\rho';t')\}$.

Note that a round is influential w.r.t. realization ρ if and only if it is (b,ρ)-influential for some b. The central property in our characterization is that each (b,ρ)-influential round is (b,ρ)-secured.

Definition 3.5.6. A deterministic allocation is called *weakly separated* if for every realization ρ and each bid vector b, it holds that if round t is $(b;\rho)$-influential with influenced agent i then it is $(b;\rho)$-secured from i.

We notice that exploration-separated is a stronger notion.

Observation 3.5.7. *For a deterministic allocation, exploration-separated implies weakly separated.*[7]

[6] Note that realizations ρ and ρ' are interchangeable.

[7] To see this, simply use the definitions. Fix realization ρ and bid vector b, let t be a $(b;\rho)$-influential round with influenced agent i. We need to show that t is $(b;\rho)$-secured from i. Round t is $(b;\rho)$-influential, thus influential w.r.t. ρ, thus (since the allocation is exploration-separated) it is bid-independent w.r.t. ρ, thus agent i cannot change allocation in round t by increasing her bid.

We are now ready to state our general characterization.

Theorem 3.5.8. *Consider the MAB mechanism design problem. Let $\mathcal{A}$ be a non-degenerate deterministic allocation rule. Then mechanism $(\mathcal{A}, \mathcal{P})$ is normalized and truthful for some payment rule $\mathcal{P}$ if and only if $\mathcal{A}$ is pointwise monotone and weakly separated.*

Proof. For the "only if" direction, $\mathcal{A}$ is pointwise-monotone by Lemma 3.5.2, and the fact that $\mathcal{A}$ is weakly separated (for k agents) is proved next in Lemma 3.5.9 similarly to Proposition 3.5.3 (albeit with a few extra details).

Lemma 3.5.9. *Consider the MAB mechanism design problem. Let $(\mathcal{A}, \mathcal{P})$ be a normalized truthful mechanism such that $\mathcal{A}$ is a non-degenerate deterministic allocation rule. Then $\mathcal{A}$ is weakly separated.*

Proof. Assume $\mathcal{A}$ is not weakly separated. Then there is a *counterexample* (ρ, b, t, t', i): a realization ρ, bid vector b, rounds t, t' and agent i such that round t is $(b;\rho)$-influential with influenced agent i and influenced round t' and it does not holds that round t is $(b;\rho)$-secured from i. We prove that this leads to a contradiction..

Let us pick a counterexample (ρ, b, t, t', i) with a minimum value of t' over all choices of (ρ, b, t, i). Without loss of generality, let us assume that $\rho_i(t') = 1$ and $\rho_j(t'') = 0$ for all $t'' > t'$ and for all agents j.

Let $j = \mathcal{A}(b;\rho;t)$. As it does not holds that round t is $(b;\rho)$-secured from i, this means that $j \neq i$, and there exists a bid $b_i^+ > b_i$ such that $\mathcal{A}(b_i^+, b_{-i};\rho;t) \neq j$.

Let $\rho' = \rho \oplus \mathbf{1}(j,t)$. The mechanism needs to compute prices for agent i when her bid is b_i^+ under realizations ρ and ρ', that is to compute $\mathcal{P}_i(b_i^+, b_{-i};\rho)$ and

$\mathcal{P}_i(b_i^+, b_{-i}; \rho')$. Therefore, the mechanism needs to compute the integral $I_i(\varphi) = \int_{x \le b_1^+} \mathcal{C}_i(x, b_{-i}; \varphi)\, dx$ for both realizations $\varphi \in \{\rho, \rho'\}$.

First of all, for all $x \le b_i^+$ and for all $t'' < t'$, $\mathcal{A}_i(x, b_{-i}; \rho; t'') = \mathcal{A}_i(x, b_{-i}; \rho'; t'')$. If not,then the minimality of t' will be violated. This is because, if there were such an x and $t'' < t'$ with $\mathcal{A}_i(x, b_{-i}; \rho; t'') \neq \mathcal{A}_i(x, b_{-i}; \rho'; t'')$, then round t will still be (b, ρ)-influential with influenced agent i, and influenced round $t'' < t'$, violating the minimality of t''. Therefore, when we decrease the bid of agent i, the only difference in the allocation can occur at time round t'.

As i is the influenced agent at round t' it must hold that $\mathcal{A}_i(b_i, b_{-i}; \rho; t') \neq \mathcal{A}_i(b_i, b_{-i}; \rho', t')$. Let us assume $0 = \mathcal{A}_i(b_i, b_{-i}; \rho; t') < \mathcal{A}_i(b_i, b_{-i}; \rho', t') = 1$ (otherwise, we can swap ρ and ρ'). Note that we have made use of the fact that the mechanism is deterministic. Let us make the the claim that for all bids $x \le b_i^+$ the influence of round t on round t' is in the same "direction."

$$\mathcal{A}_i(x, b_{-i}; \rho; t') \le \mathcal{A}_i(x, b_{-i}; \rho'; t') \text{ for all } x \le b_i^+. \tag{3.5.3}$$

Suppose (3.5.3) does not hold. Then there is an $x \le b_i^+$ such that $1 = \mathcal{A}_i(x, b_{-i}; \rho; t') > \mathcal{A}_i(x, b_{-i}; \rho'; t') = 0$. (Note that we have used the fact that the mechanism is deterministic.) If $x > b_i$, then pointwise monotonicity is violated in ρ', since $0 = \mathcal{A}_i(x, b_{-i}; \rho'; t') < \mathcal{A}_i(b_i, b_{-i}; \rho'; t') = 1$. If $x < b_i$ on the other hand, then the pointwise-monotonicity is violated in ρ, since $1 = \mathcal{A}_i(x, b_{-i}; \rho; t') > \mathcal{A}_i(b_i, b_{-i}; \rho; t') = 0$, giving a contradiction in both cases. The claim (3.5.3) follows.

By the non-degeneracy of $\mathcal{A}$, there exists a non-degenerate interval I containing b_i such that

$$\mathcal{A}_i(x, b_{-i}; \rho; t') < \mathcal{A}_i(x, b_{-i}; \rho'; t') \text{ for all } x \in I. \tag{3.5.4}$$

By (3.5.3) and (3.5.4) it follows that $I_i(\rho) < I_i(\rho')$. However, the mechanism

cannot distinguish between ρ and ρ' when agent i's bid is b_i^+, since the differing bit $\rho_j(t)$ is not seen. Contradiction. $\square$

We now continue the proof of Theorem 3.5.8 and focus on its "if" direction ($\Leftarrow$ direction). Let $\mathcal{A}$ be a deterministic allocation rule which is pointwise monotone and weakly separated. We need to provide a payment rule $\mathcal{P}$ such that the resulting mechanism $(\mathcal{A}, \mathcal{P})$ is truthful and normalized. Since $\mathcal{A}$ is pointwise monotone, it immediately follows that it is monotone (i.e., as an agent increases her bid, the number of clicks that she gets cannot decrease). Therefore it follows from Theorem 3.5.1 that mechanism $(\mathcal{A}, \mathcal{P})$ is truthful and normalized if and only if $\mathcal{P}$ is given by (3.5.1). We need to show that $\mathcal{P}$ can be computed using only the knowledge of the clicks (bits from the realization) that were revealed during the execution of $\mathcal{A}$.

Assume we want to compute the payment for agent i in bid profile (b_i, b_{-i}) and realization ρ. We will prove that we can compute $\mathcal{C}_i(x) := \mathcal{C}_i(x, b_{-i}; \rho)$ for all $x \leq b_i$. To compute $\mathcal{C}_i(x)$, we show that it is possible to simulate the execution of the mechanism with $\texttt{bid}_i = x$. In some rounds, the agent i loses an impression, and in others it retains the impression (pointwise monotonicity ensures that agent i cannot gain an impression when decreasing her bid). In rounds that it loses an impression, the mechanism does not observe the bits of ρ in those rounds, so we prove that those bits are *irrelevant* while computing $\mathcal{C}_i(x)$. In other words, while running with $\texttt{bid}_i = x$, if mechanism needs to observe the bit that was not revealed when running with $\texttt{bid}_i = b_i$, we arbitrarily put that bit equal to 1 and simulate the execution of $\mathcal{A}$. We want to prove that this computes $\mathcal{C}_i(x)$ correctly.

Let $t_1 < t_2 < \cdots < t_n$ be the rounds in which agent i did not get an impression while bidding x, but did get an impression while bidding b_i. Let $\rho^0 := \rho$, and let

us define realization ρ^l inductively for every $l \in [n]$ by setting $\rho^l := \rho^{l-1} \oplus \mathbf{1}(j_l,\, t_l)$, where $j_l = \mathcal{A}(x, b_{-i}; \rho^{l-1}; t_l)$ is the agent that got the impression at round t_l with realization ρ^{l-1} and bids (x, b_{-i}).

First, we claim that $j_l \neq i$ for any l. Indeed, suppose not, and pick the smallest l such that $j_{l+1} = i$. Then t_l is a $(x, b_{-i};\, \rho^l)$-influential round, with influenced agent $j_{l+1} = i$. Thus t_l is $(x, b_{-i};\, \rho^l)$-secured from i. Since $\mathcal{A}(x, b_{-i};\, \rho^l; t_l) = \mathcal{A}(x, b_{-i};\, \rho^{l-1}; t_l) = j_l \neq i$ by minimality of l, agent i does not get an impression in round t_l if she raises her bid to b_i. That is, $\mathcal{A}(b; \rho^l; t_l) \neq i$. However, the changes in realizations $\rho^0\,,\ \ldots\ , \rho^{l-1}$ only concern the rounds in which agent i is chosen, so they are not seen by the allocation if the bid profile is b (to prove this formally, use induction). Thus, $\mathcal{A}(b; \rho^l; t_l) = \mathcal{A}(b; \rho; t_l) = i$, contradiction. Claim proved. It follows that $\mathcal{A}(b; \rho; t_l) = i$ for each l. (This is because by induction, the change from ρ^{l-1} to ρ^l is not seen by the allocation if the bid profile is b.)

We claim that $\mathcal{A}_i(x, b_{-i}; \rho; t') = \mathcal{A}_i(x, b_{-i}; \rho^n; t')$ for every round t', which will prove the theorem. If not, then there exists l such that $\mathcal{A}_i(x, b_{-i}; \rho^l; t') \neq \mathcal{A}_i(x, b_{-i}; \rho^{l-1}; t')$ for some t' (and of course $t' > t_l$). Round t_l is thus $(x, b_{-i}; \rho^l)$-influential with influenced round t' and influenced agent i. Moreover, the influencing agent of that round is j_l, and we already proved that $j_l \neq i$. Since round t_l is $(x, b_{-i}; \rho^l)$-secured from agent i due to the "weakly separated" condition, it follows that agent i does not get an impression in round t_l if she raises her bid to b_i. That is, $\mathcal{A}(b; \rho^l; t_l) \neq i$, contradiction.

This finishes the proof of Theorem 3.5.8. □

Note that we have proven the main characterization (Theorem 3.2.3) for the case of two agents, because for two agents IIA trivially holds and in the scale-free

case, an allocation is exploration-separated if and only if it is weakly separated.

Necessity of non-degenerate assumption Let us argue that the non-degeneracy assumption in Theorem 3.5.8 is indeed necessary. To this end, let us present a simple deterministic mechanism $(\mathcal{A}, \mathcal{P})$ for two agents that is truthful and normalized, such that the allocation rule $\mathcal{A}$ is pointwise monotone, scale-free and yet *not* weakly separated. (The catch is, of course, that it is degenerate.) There are only two rounds. Agent 1 allocated at round 1 if and only if $b_1 \geq b_2$. Agent 1 allocated at round 2 if $b_1 > b_2$ or if $b_1 = b_2$ and $\rho_1(1) = 1$; otherwise agent 2 is shown. This completes the description of the allocation rule. To obtain a payment rule $\mathcal{P}$ which makes the mechanism normalized and truthful, consider an alternate allocation rule $\mathcal{A}'$ which in each round selects agent 1 if and only if $b_1 \geq b_2$. (Note that $\mathcal{A}' = \mathcal{A}$ except when $b_1 = b_2$.) Use Theorem 3.5.8 for $\mathcal{A}'$ to obtain a normalized truthful mechanism $(\mathcal{A}', \mathcal{P}')$, and set $\mathcal{P} = \mathcal{P}'$. The payment rule $\mathcal{P}$ is well-defined since the observed clicks for $\mathcal{P}$ and $\mathcal{P}'$ coincide unless $b_1 = b_2$, in which case both payment rules charge 0 to both players. The resulting mechanism $(\mathcal{A}, \mathcal{P})$ is normalized and truthful because the integral in (3.5.1) remains the same even if we change the value at a single point. It is easy to see that the allocation rule $\mathcal{A}$ has all the claimed properties; it fails to be non-degenerate because round t is influential only when $b_1 = b_2$.

3.5.2 Scalefree and IIA allocation rules

To complete the proof of Theorem 3.2.3, we show that under the right assumptions, an allocation is exploration-separated if and only if it is weakly separated.

Lemma 3.5.10. *Consider the MAB mechanism design problem. Let $\mathcal{A}$ be a non-degenerate deterministic allocation rule which is scalefree, pointwise monotone, and*

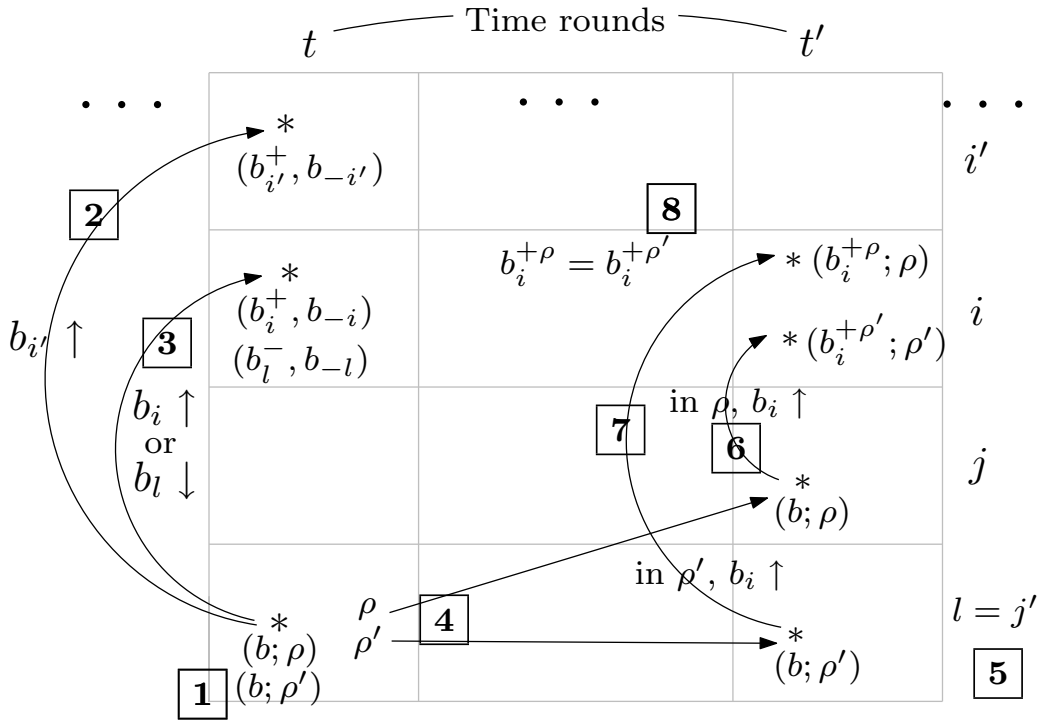

Figure 3.1: This figure explains all the steps in the proof of Lemma 3.5.10. The rows correspond to agents (whose identity is shown on the right side), and columns correspond to time rounds. The asterisks show the impressions. The arrows show how the impressions get *transferred*, and labels on the arrows show what causes the transfer. In labels, "in ρ, $b_i \uparrow$" denotes that a particular transfer of impression is caused in realization ρ when bid b_i in increased.

satisfies IIA. Then it is exploration-separated if and only if it is weakly separated.

Before presenting the full proof of the lemma, we present the proof sketch with gives the main ideas.

Proof sketch of Theorem 3.5.10. We sketch the proof of Lemma 3.5.10 at a *very* high level. The "only if" direction ($\Rightarrow$ direction) was observed in Observation 3.5.7. For the "if" direction, let $\mathcal{A}$ be a weakly-separated mechanism. We prove by a contradiction that it is exploration-separated. If not, then there is a realization ρ

and a round t such that t is influencial w.r.t. ρ as well as not bid-dependent w.r.t. ρ. Let round t be influencial with bid vector b, influencing agent l, and influenced agents j and $j' \neq j$ in influenced round t' (see 1 in Figure 3.1; all boxed numbers in this sketch will refer to this figure).

From the assumption, t is not bid-dependent w.r.t. ρ, which means that there exists a bid profile b' such that $i' \neq l$ is played in round t with bids b'. Using scalefreeness, IIA, and pointwise-monotonicity, we can prove that there exists a sufficiently large bid $b_{i'}^{+}$ of agent i' such that she gets an impression in round t with bids $(b_{i'}^{+}, b_{-i'})$ (see 2). Using the properties of the mechanism, it can further be proved that there is an agent i such that she gets the impression in round t when either i increases her bid, *or* l decreases her bid (see 3). When i increases her bid to b_i^{+}, she also gets an impression in round t', since impressions cannot differ in round t' in the case when l is not played in round t and they must get transferred from j and j' to *somebody* in round t', and IIA implies that this *somebody* should be i.

Recall that two different players j and j' get the impression in round t' under ρ and ρ' respectively (see 4). We prove that either agent j' or agent j must be equal to l (this is done by looking at how the allocation in round t' changes when l decreases her bid). Let us break the symmetry and assume $j' = l$ (see box 5). It is also easy to see that when i increases her bid, impression in round t' get transferred to her in ρ (at some minimum value $b_i^{+\rho}$, see 6), and impression in round t' gets transferred to her also in ρ' (as some possibly different minimum value $b_i^{+\rho'}$, see 7). Using the assumptions of weakly-separatedness, we prove that $b_i^{+\rho} = b_i^{+\rho'}$ (see 8). This can be proved by observing that $b_i^{+} \geq \max\{b_i^{+\rho}, b_i^{+\rho'}\}$, and then using weakly-separatedness of $\mathcal{A}$. Since these two bids were at a "threshold value" (these

were the minimum values of bids to have transferred the impression in ρ and ρ' from j and l respectively), we are able to prove that the ratio of b_j/b_l must be some fixed number dependent on ρ, ρ', and t'. In particular, it follows that b_l belongs to a finite set $S(b_{-l})$ which depends only on b_{-l}. However, by non-degeneracy of $\mathcal{A}$ there must be infinitely many such b_l's, which leads to a contradiction. □

In the rest of this section, we present the full proof of the "if" direction of Lemma 3.5.10.

For bid profile b, realization ρ, agent l and round t, the tuple $(b; \rho; l; t)$ is called an *influence-tuple* if round t is (b, ρ)-influential with influencing agent l. Suppose allocation $\mathcal{A}$ is weakly separated but not exploration-separated. Then there is a *counterexample*: an influence-tuple $(b; \rho; l; t)$ such that round t is not bid-independent w.r.t. realization ρ. We prove that such counterexample can occur only if $b_l \in S_l(b_{-l})$, for some finite set $S_l(b_{-l}) \subset \mathbb{R}$ that depends only on b_{-l}.

Proposition 3.5.11. *Let $\mathcal{A}$ be as in Lemma 3.5.10. Assume $\mathcal{A}$ is weakly separated. Then for each agent l and each bid profile b_{-l} there exists a finite set $S_l(b_{-l}) \subset \mathbb{R}$ with the following property: for each counterexample $(b_l, b_{-l}; \rho; l; t)$ it is the case that $b_l \in S_l(b_{-l})$.*

Once this proposition is proved, we obtain a contradiction with the non-degeneracy of $\mathcal{A}$. Indeed, suppose $(b; \rho; l; t)$ is a counterexample. Then $(b; \rho; l; t)$ is an influence-tuple. Since $\mathcal{A}$ is non-degenerate, there exists a non-degenerate interval I such that for each $x \in I$ it holds that $(x, b_{-l}; \rho; l; t)$ is an influence-tuple, and therefore a counterexample. Thus the set $S_l(b_{-l})$ in Proposition 3.5.11 cannot be finite, contradiction.

In the rest of this section we prove Proposition 3.5.11. Fix a counterexample

$(b;\rho;l;t)$; let $t' > t$ be the influenced round. In particular, $\mathcal{A}(b;\rho;t) = l$ (see $\boxed{1}$ in Figure 3.1 on page 70; all boxed numbers will refer to this figure). Then by the assumption there exist bids b' such that $\mathcal{A}(b';\rho;t) = i' \neq l$. We claim that this implies that there exists a bid $b^+_{i'} > b_{i'}$ such that $\mathcal{A}(b^+_{i'}, b_{-i'};\rho;t) = i'$ (see $\boxed{2}$). This is proven in Lemma 3.5.13 below, and in order to prove it we first present the following lemma, which essentially states that if the mechanism makes a choice between i and j of who to be show, then it can only depend on the ratio of their bids $\texttt{bid}_i/\texttt{bid}_j$, and not on the bids of other agents.

Lemma 3.5.12. *Let $\mathcal{A}$ be an MAB (deterministic) allocation rule that is pointwise-monotone, scalefree, and satisfies IIA. Let there be two bid profiles α and β such that $\mathcal{A}(\alpha;\rho;t) \in \{i,j\}$, $\mathcal{A}(\beta;\rho;t) \in \{i,j\}$, and $\alpha_i/\alpha_j = \beta_i/\beta_j$. Then it must be the case that $\mathcal{A}(\alpha;\rho;t) = \mathcal{A}(\beta;\rho;t)$.*

Proof. As $\mathcal{A}$ is scalefree we assume that $\alpha_i = \beta_i$ and $\alpha_j = \beta_j$ by scaling bids in β by a factor of α_i/β_i (or a factor of α_j/β_j), without changing the allocation.

Assume for the sake of a contradiction that $\mathcal{A}(\beta;\rho;t) \neq \mathcal{A}(\alpha;\rho;t)$. Let us number the agents as follows. Agents i and j are numbered 1 and 2, respectively. The rest of the agents are arbitrarily numbered 3 to k. Consider the following sequence of bid vectors. $\alpha(1) = \alpha(2) = \alpha$ and $\alpha(m) = (\beta_m, \alpha(m-1)_{-m})$ for $m \in \{3,\ldots,k\}$. As $\alpha(1) = \alpha$ and $\alpha(k) = \beta$, $\mathcal{A}(\alpha(1);\rho;t) = \mathcal{A}(\alpha;\rho;t)$ and $\mathcal{A}(\alpha(k);\rho;t) = \mathcal{A}(\beta;\rho;t)$. Since $\mathcal{A}(\alpha(k);\rho;t) = \mathcal{A}(\beta;\rho;t) \neq \mathcal{A}(\alpha;\rho;t) = \mathcal{A}(\alpha(1);\rho;t)$ there exists $m \in \{3,\ldots,k\}$ such that $\mathcal{A}(\alpha(m-1);\rho;t) = \mathcal{A}(\alpha;\rho;t) \in \{i,j\}$ while $\mathcal{A}(\alpha(m);\rho;t) \neq \mathcal{A}(\alpha(m-1);\rho;t)$. As $m \neq i$ and $m \neq j$, IIA implies that $\mathcal{A}(\alpha(m);\rho;t) = m$ and given that, IIA also implies that $\mathcal{A}(\alpha(k);\rho;t) \in \{m, m+1, \ldots k\}$ (note that i,j are not in this set). But as $\mathcal{A}(\alpha(k);\rho;t) = \mathcal{A}(\beta;\rho;t) \in \{i,j\}$ this yields a contradiction. □

Lemma 3.5.13. *Let $\mathcal{A}$ be an MAB (deterministic) allocation rule that is pointwise-monotone, scalefree, and satisfies IIA. Let there be two bid profiles α and β such that $\mathcal{A}(\alpha;\rho;t) = i$ and $\mathcal{A}(\beta;\rho;t) = j \neq i$. Then there exists $\beta_i^+ > \beta_i$ such that $\mathcal{A}(\beta_i^+, \beta_{-i};\rho;t) = i$.*

In other words, if it is possible for i to get the impression in round t at all, then it is possible for her to get the impression starting from any bid profile and raising her bid high enough.

Proof. We first note that $\frac{\alpha_i}{\alpha_j} \geq \frac{\beta_i}{\beta_j}$. If not, then $\frac{\alpha_i}{\alpha_j} < \frac{\beta_i}{\beta_j}$. Consider a raised bid of i from α_i to $\alpha_i^+ = \alpha_j \cdot \frac{\beta_i}{\beta_j}$. In the bid profile $(\alpha_i^+, \alpha_{-i})$, i must get the impression (by pointwise monotonicity). This gives a contradiction to Lemma 3.5.12, since $\mathcal{A}(\alpha_i^+, \alpha_{-i};\rho;t) = i \in \{i,j\}$, $\mathcal{A}(\beta;\rho;t) = j \in \{i,j\}$, and $\frac{\alpha_i^+}{\alpha_j} = \frac{\beta_i}{\beta_j}$, but $\mathcal{A}(\alpha_i^+, \alpha_{-i};\rho;t) \neq \mathcal{A}(\beta;\rho;t)$.

Now, consider i increasing her bid in profile β to $\beta_i^+ = \beta_j \cdot \frac{\alpha_i}{\alpha_j}$. Now, $\mathcal{A}(\alpha;\rho;t) = i \in \{i,j\}$, $\mathcal{A}(\beta_i^+, \beta_{-i};\rho;t) \in \{i,j\}$ (from IIA), and $\frac{\alpha_i}{\alpha_j} = \frac{\beta_i^+}{\beta_j}$. We can apply Lemma 3.5.12 to deduce that $\mathcal{A}(\alpha;\rho;t) = \mathcal{A}(\beta_i^+, \beta_{-i};\rho;t)$ and both are equal to i since the first allocation is equal to i. □

From the lemma above, it follows that agent i' can increase her bid (in bid profile b) and get the impression in realization ρ, round t. To quantify by how much agent i' needs to raise her bid to get the impression, we introduce the notion of *threshold* $\Theta_{i,j}(\rho;t)$ in the next lemma.

Lemma 3.5.14. *Let $\mathcal{A}$ be an MAB (deterministic) allocation rule that is pointwise monotone, scalefree and satisfies IIA. For realization ρ, round t, two agents i and $j \neq i$, let bids b_{-i-j} be such that there exist x_0 and y satisfying $\mathcal{A}(x_0, y, b_{-i-j};\rho;t) = j$, and there exists x (possibly dependent on y) satisfying $\mathcal{A}(x, y, b_{-i-j};\rho;t) = i$.*

Let us fix such a y and define[8]

$$\Theta_{i,j}^{b_{-i-j}}(\rho,t) = \tfrac{1}{y}\inf_x \left\{x \;\middle|\; \mathcal{A}(x,y,b_{-i};\rho;t) = i\right\}.$$

Then for any bids b'_{-i-j}, $\Theta_{i,j}^{b'_{-i-j}}(\rho,t)$ is well defined and satisfies $\Theta_{i,j}^{b'_{-i-j}}(\rho,t) = \Theta_{i,j}^{b_{-i-j}}(\rho,t)$. We denote it by $\Theta_{i,j}(\rho,t)$, as $\Theta_{i,j}^{b_{-i-j}}(\rho,t)$ is independent of b_{-i-j}.

Proof. We first prove that if the conditions of the definition of $\Theta_{i,j}^{b_{-i-j}}(\rho;t)$ are satisfied for b_{-i-j}, then are also satisfied for any other b'_{-i-j}. Let us say they are satisfied for b_{-i-j}, that is there exists x_0, x and y, such that $\mathcal{A}(x_0,y,b_{-i-j};\rho;t) = j$ and $\mathcal{A}(x,y,b_{-i};\rho;t) = i$. We want to prove existence of x' and y' for b'_{-i-j}. If $\mathcal{A}(x_0,y,b'_{-i-j};\rho;t) = j$ then existence of y' is proved for b'_{-i-j} too, since $y' = y$ works. If not, then $\mathcal{A}(x_0,y,b'_{-i-j};\rho;t) = j' \neq j$ and $\mathcal{A}(x_0,y,b_{-i-j};\rho;t) = j$, and by Lemma 3.5.13, there exists a $y' > y$ such that $\mathcal{A}(x_0,y',b'_{-i-j};\rho;t) = j$. Once the existence of y' is proved, we now prove the existence of x'. Let $x' = x \cdot \frac{y'}{y} \geq x$. We have $\mathcal{A}(x,y,b_{-i-j};\rho;t) = i \in \{i,j\}$ and $\mathcal{A}(x',y',b'_{-i-j};\rho;t) \in \{i,j\}$ by IIA (i can only transfer impression to her by changing her bid) and $x'/y' = x/y$. From Lemma 3.5.12, we get $i = \mathcal{A}(x,y,b_{-i-j};\rho;t) = \mathcal{A}(x',y',b'_{-i-j};\rho;t)$. Hence the existence of x' is proved too.

For the sake of contradiction, let us assume that $\theta := \Theta_{i,j}^{b_{-i-j}}(\rho;t) < \Theta_{i,j}^{b'_{-i-j}}(\rho;t) =: \theta'$. Let us scale the bids in (x',y',b'_{-i-j}) by a factor such that the factor times y' is equal to y. We can hence assume that $y' = y$. Let us pick a bid $x'' \in (\theta y, \theta' y)$. We have $\mathcal{A}(x'',y,b_{-i-j};\rho;t) = i$ (since x''/y is past the threshold θ), $\mathcal{A}(x'',y' = y,b'_{-i-j};\rho;t) = j$ (x''/y' is yet not past the threshold θ'), and

[8]Note that if there are no values of bids of i (x_0 and x) and j (equal to y) such that j can get an impression with small enough bid (x_0) of agent i *and* i can get an impression by raising her bid (to x), then we don't define $\Theta_{i,j}^{b_{-i-j}}(\rho;t)$ at all. We will be careful not to use such undefined Θ's. It is not hard to see that if bids are nonzero, then $\Theta_{i,j}(\rho;t)$ is defined if and only if $\Theta_{j,i}(\rho;t)$ is. Moreover $0 < \Theta_{i,j}(\rho;t) < \infty$, and $\Theta_{j,i}(\rho;t) = (\Theta_{i,j}(\rho;t))^{-1}$.

$x''/y = x''/y'$. This is a contradiction to the Lemma 3.5.12. Therefore, $\theta = \theta'$. □

We conclude that if $b^+_{i'} > b_l \cdot \Theta_{i',l}(\rho, t)$ then $\mathcal{A}(b^+_{i'}, b_{-i'}; \rho; t) = i' \neq l$ (see 2 again). Note that we are using $\Theta_{i',l}(\rho; t)$ since this is well-defined. Define $\rho' = \rho \oplus \mathbf{1}(l, t)$.

Let us think about decreasing the bid of agent l from b_l (it is positive, since all bids are assumed to be positive). When the bid of agent l is b_l, she gets the impression in round t, but when her bid is small enough (in particular as low as $b_{i'}/\Theta_{i',l}(\rho; t)$), then she must not get the impression in round t (see Lemma 3.5.12). When the bid of l decreases, some other agent gets the impression in round t, let us call that agent i (note that this agent may not be the same as agent i' above). See 3.

Now, starting from bid profile b, let us increase the bid of agent i. When the bid of agent i is large enough (in particular as large as $b_i \Theta_{i',l}(\rho; t) b_l / b_{i'}$), then l can no longer get the impression in round t (see Lemma 3.5.12). From IIA, the impression must get transferred to i. Therefore we can define $\Theta_{i,l}(\rho; t)$, and when $b^+_i > b_l \Theta_{i,l}(\rho; t)$, agent i gets the impression in round t (see 3 again). Note that $\mathcal{A}(b^+_i, b_{-i}; \rho; t) = \mathcal{A}(b^+_i, b_{-i}; \rho'; t) = i$ (click information for l at round t cannot influence the impression decision at round t).

Recall that t' is the influenced round. Let $\mathcal{A}(b; \rho; t') = j$ and let $\mathcal{A}(b; \rho'; t') = j' \neq j$ (see 4). As $\mathcal{A}$ is pointwise monotone and IIA, $\mathcal{A}(b^+_i, b_{-i}; \rho; t') \in \{i, j\}$ and $\mathcal{A}(b^+_i, b_{-i}; \rho'; t') \in \{i, j'\}$. It must be the case that $\mathcal{A}(b^+_i, b_{-i}; \rho; t') = \mathcal{A}(b^+_i, b_{-i}; \rho'; t')$, as l does not get an impression at round t (and the algorithm does not see the difference between ρ and ρ'). As $j' \neq j$ we conclude that

$$\mathcal{A}(b^+_i, b_{-i}; \rho; t') = \mathcal{A}(b^+_i, b_{-i}; \rho'; t') = i.$$

Next we note that $i \neq j$ and $i \neq j'$. This is because if $i = j$ (respectively $i = j'$), then round t would be $(b;\rho)$-influential (respectively $(b;\rho')$-influential) with influenced agent i but it is not $(b;\rho)$-secured (respectively $(b;\rho')$-secured) from i, in contradiction to the assumption.

We also note that $l \in \{j, j'\}$ (see 5). Assume for the sake of contradiction that $l \neq j$ and $l \neq j'$. For $b_l^- < b_i \cdot \Theta_{l,i}(\rho, t)$ it holds that $\mathcal{A}(b_l^-, b_{-l}; \rho; t) = \mathcal{A}(b_l^-, b_{-l}; \rho'; t) = i$ (since i was defined such that i gets the impression in round t when l decreases her bid) thus $\mathcal{A}(b_l^-, b_{-l}; \rho; t') = \mathcal{A}(b_l^-, b_{-l}; \rho'; t')$ (as click information for l at round t is not observed). (Also, as a side note, observe that $b_l^- < b_l$ by pointwise-monotonicity since agent l was getting an impression in round t with bid b_l and lost it when her bid is b_l^-.) Let $\mathcal{A}(b_l^-, b_{-l}; \rho; t') = \mathcal{A}(b_l^-, b_{-l}; \rho'; t') = l'$. Note that $l' \neq l$, since otherwise, $\mathcal{A}_l(x, b_{-l}; \rho; t')$ is not a monotone function of x: it is 0 when $x = b_l$ (since j gets an impression), and 1 when $x = b_l^- < b_l$, a contradiction to pointwise-monotonicity. Now, note that the impression in ρ' at time t' transfers from j' to l', and impression in ρ at time t' transfers from j to l', none of which ($\{j, j', l'\}$) are equal to l and $j \neq j'$. Let us write this in equations:

$$\mathcal{A}(b_l, b_{-l}; \rho; t') = j \qquad \mathcal{A}(b_l^-, b_{-l}; \rho; t') = l'$$
$$\mathcal{A}(b_l, b_{-l}; \rho'; t') = j' \qquad \mathcal{A}(b_l^-, b_{-l}; \rho'; t') = l'.$$

It must be the case that either $j \neq l'$ or $j' \neq l'$ (since $j \neq j'$). If $j \neq l'$, then in ρ at time t', reducing the bid of l transfers impression from j to l' (both of them are different from l), thus violating IIA. Similarly, if $j' \neq l'$, then in ρ' at time t', reducing the bid of l transfers impression from j' to l' (both of them are different from l), thus violating IIA. We thus have $l \in \{j, j'\}$. Let $l = j'$ (since otherwise, we can swap the roles of ρ and ρ').

To summarize what we have proved so far: there are 3 distinct agents i, j, l

such that

$$\mathcal{A}(b;\rho;t) = \mathcal{A}(b;\rho';t) = \mathcal{A}(b;\rho';t') = l \quad (\text{since } \mathcal{A}(b;\rho';t') = j' = l),$$

$$\mathcal{A}(b;\rho;t') = j \quad \text{and}$$

$$\mathcal{A}(b_i^+, b_{-i};\rho;t) = \mathcal{A}(b_i^+, b_{-i};\rho;t') = \mathcal{A}(b_i^+, b_{-i};\rho';t) = \mathcal{A}(b_i^+, b_{-i};\rho';t') = i.$$

Observe also that $\Theta_{i,l}(\rho,t) = \Theta_{i,l}(\rho',t)$ as ρ and ρ' only differ at a click at round t, and such a click cannot determine the allocation decision at round t. Also, $\max\{\Theta_{i,j}(\rho,t')\cdot b_j, \Theta_{i,l}(\rho',t')\cdot b_l\} \leq \Theta_{i,l}(\rho,t)\cdot b_l$ as the allocation at round t', which is different for ρ and ρ' (at b), depends on l getting the impression at round t.[9] Finally we prove that $\Theta_{i,j}(\rho,t')\cdot b_j = \Theta_{i,l}(\rho',t')\cdot b_l$ (see 8).

Claim 3.5.15. $\Theta_{i,j}(\rho,t')\cdot b_j = \Theta_{i,l}(\rho',t')\cdot b_l$

Proof. First of all, note that $\Theta_{i,j}(\rho;t')$ and $\Theta_{i,l}(\rho',t')$ are well-defined. Let $\bar{b}_i = (\Theta_{i,j}(\rho,t')\cdot b_j + \Theta_{i,l}(\rho',t')\cdot b_l)/2$. Consider the following two cases.

If $\Theta_{i,j}(\rho,t')\cdot b_j < \Theta_{i,l}(\rho',t')\cdot b_l$ then round t is $(\bar{b}_i, b_{-i};\rho)$-influential (as $\mathcal{A}(\bar{b}_i, b_{-i};\rho;t') = i$ and $\mathcal{A}(\bar{b}_i, b_{-i};\rho';t') = l$) with influencing agent l ($\mathcal{A}(\bar{b}_i, b_{-i};\rho;t) = \mathcal{A}(\bar{b}_i, b_{-i};\rho';t) = l$ since $\bar{b}_i < \Theta_{i,l}(\rho,t)\cdot b_l$) and influenced agent i. Additionally, t it is not $(\bar{b}_i, b_{-i};\rho)$-secured from i (as $\mathcal{A}(b_i^+, b_{-i};\rho;t) = \mathcal{A}(b_i^+, b_{-i};\rho';t) = i$). A contradiction to first condition in the theorem.

Similarly, if $\Theta_{i,j}(\rho,t')\cdot b_j > \Theta_{i,l}(\rho',t')\cdot b_l$ then round t is $(\bar{b}_i, b_{-i};\rho)$-influential (as now $\mathcal{A}(\bar{b}_i, b_{-i};\rho;t') = j$ and $\mathcal{A}(\bar{b}_i, b_{-i};\rho';t') = i$) with influencing agent l and influenced agent i. Additionally, t it is not $(\bar{b}_i, b_{-i};\rho)$-secured from i. Again, a contradiction to the first condition in the theorem. □

[9] In Figure 3.1 we defined $b_i^{+\rho} := \Theta_{i,j}(\rho;t')b_j$ and $b_i^{+\rho'} := \Theta_{i,l}(\rho';t')b_l$. These are the bids of agent i at which impression transfers to her in round t' in ρ and ρ' respectively. See 6 and 7 in the figure.

The lemma implies that $b_l \in S_l(b_{-l})$, where a finite set $S_l(b_{-l})$ is defined by

$$S_l(b_{-l}) = \left\{ b_j \, \frac{\Theta_{i,j}(\rho, t')}{\Theta_{i,l}(\rho', t')} : \quad \begin{array}{l} \text{all agents } i, j \neq l, \text{ all realizations } \rho, \rho' \\ \text{and all } t' \text{ s.t. } \dfrac{\Theta_{i,j}(\rho, t')}{\Theta_{i,l}(\rho', t')} \text{ is well-defined} \end{array} \right\}.$$

This completes the proof of Proposition 3.5.11.

3.6 Lower bounds on regret

In this section we use structural results from the previous section to derive lower bounds on regret.

Theorem 3.6.1. *Consider the stochastic MAB mechanism design problem with k agents. Let $\mathcal{A}$ be an exploration-separated deterministic allocation rule. Then its regret is $R(T; v_{\max}) = \Omega(v_{\max} k^{1/3} T^{2/3})$.*

Let $\vec{\mu}_0 = (\frac{1}{2}, \ \ldots \ , \frac{1}{2}) \in [0,1]^k$ be the vector of CTRs in which for each agent the CTR is $\frac{1}{2}$. For each agent i, let $\vec{\mu}_i = (\mu_{i1}, \ldots, \mu_{ik}) \in [0,1]^k$ be the vector of CTRs in which agent i has CTR $\mu_{ii} = \frac{1}{2} + \epsilon$, $\epsilon = k^{1/3} T^{-1/3}$, and every other agent $j \neq i$ has CTR $\mu_{ij} = \frac{1}{2}$. As a notational convention, denote by $\mathbb{P}_i[\cdot]$ and $\mathbb{E}_i[\cdot]$ respectively the probability and expectation induced by the algorithm when clicks are given by $\vec{\mu}_i$. Let $\mathcal{I}_i$ be the problem instance in which CTRs are given by $\vec{\mu}_i$ and all bids are $v_{\max}$. For each agent i, let $\mathcal{J}_i$ be the problem instance in which CTRs are given by $\vec{\mu}_0$, the bid of agent i is $v_{\max}$, and the bids of all other agents are $v_{\max}/2$. We will show that for any exploration-separated deterministic allocation rule $\mathcal{A}$, one of these $2k$ instances causes high regret.

Let N_i be the number of bid-independent rounds in which agent i is played. Note that N_i does not depend on the bids. It is a random variable in the probability

space induced by the clicks; its distribution is completely specified by the CTRs. We show that (in a certain sense) the allocation cannot distinguish between $\vec{\mu}_0$ and $\vec{\mu}_i$ if N_i is too small. Specifically, let $\mathcal{A}_t$ be the allocation in round t. Once the bids are fixed, this is a random variable in the probability space induced by the clicks. For a given set S of agents, we consider the event $\{\mathcal{A}_t \in S\}$ for some fixed round t, and upper-bound the difference between the probability of this event under $\vec{\mu}_0$ and $\vec{\mu}_i$ in terms of $\mathbb{E}_i[N_i]$, in the following crucial claim, which we prove at the end of this section (in Section 3.6.1).

Claim 3.6.2. *For any fixed vector of bids, each round t, each agent i and each set of agents S, we have*

$$|\,\mathbb{P}_0[\mathcal{A}_t \in S] - \mathbb{P}_i[\mathcal{A}_t \in S]\,| \leq O\left(\epsilon^2\,\mathbb{E}_0[N_i]\right). \tag{3.6.1}$$

Proof of Theorem 3.6.1: Fix a positive constant β to be specified later. Consider the case $k = 2$ first. If $\mathbb{E}_0[N_i] > \beta\,T^{2/3}$ for some agent i, then on the problem instance $\mathcal{J}_i$, regret is $\Omega(T^{2/3})$. So without loss of generality let us assume $\mathbb{E}_0[N_i] \leq \beta\,T^{2/3}$ for each agent i. Then, plugging in the values for ϵ and $\mathbb{E}_0[N_i]$, the right-hand side of (3.6.1) is at most $O(\beta)$. Take β so that the right-hand side of (3.6.1) is at most $\frac{1}{4}$. For each round t there is an agent i such that $\mathbb{P}_0[\mathcal{A}_t \neq i] \geq \frac{1}{2}$. Then $\mathbb{P}_i[\mathcal{A}_t \neq i] \geq \frac{1}{4}$ by Claim 3.6.2, and therefore in this round algorithm $\mathcal{A}$ incurs regret $\Omega(\epsilon\,v_{\max})$ under problem instance $\mathcal{I}_i$. By Pigeonhole Principle there exists an i such that this happens for at least half of the rounds t, which gives the desired lower-bound.

Case $k \geq 3$ requires a different (and somewhat more complicated) argument. Let $R = \beta\,k^{1/3}\,T^{2/3}$ and N be the number of bid-independent rounds. Assume $\mathbb{E}_0[N] > R$. Then $\mathbb{E}_0[N_i] \leq \frac{1}{k}\,\mathbb{E}_0[N]$ for some agent i. For the problem instance

$\mathcal{J}_i$ there are, in expectation, $E[N - N_i] = \Omega(R)$ bid-independent rounds in which agent i is not played; each of which contributes $\Omega(v_{\max})$ to regret, so the total regret is $\Omega(v_{\max} R)$.

From now on assume that $\mathbb{E}_0[N] \leq R$. Note that by Pigeonhole Principle, there are more than $\frac{k}{2}$ agents i such that $\mathbb{E}_0[N_i] \leq 2R/k$. Furthermore, let us say that an agent i is *good* if $\mathbb{P}_0[\mathcal{A}_t = i] \leq \frac{4}{5}$ for more than $T/6$ different rounds t. We claim that there are more than $\frac{k}{2}$ good agents. Suppose not. If agent i is not good then $\mathbb{P}_0[\mathcal{A}_t = i] > \frac{4}{5}$ for at least $\frac{5}{6}T$ different rounds t, so if there are at least $k/2$ such agents then

$$T = \sum_{t=1}^{T} \sum_{i=1}^{k} \mathbb{P}_0[\mathcal{A}_t = i] > \frac{k}{2} \times \left(\frac{5}{6}T\right) \times \frac{4}{5} \geq kT/3 \geq T,$$

contradiction. Claim proved. It follows that there exists a good agent i such that $\mathbb{E}_0[N_i] \leq 2R/k$. Therefore the right-hand side of (3.6.1) is at most $O(\beta)$. Pick β so that the right-hand side of (3.6.1) is at most $\frac{1}{10}$. Then by Claim 3.6.2 for at least $T/6$ different rounds t we have $\mathbb{P}_i[\mathcal{A}_t = i] \leq \frac{9}{10}$. In each such round, if agent i is not played then algorithm $\mathcal{A}$ incurs regret $\Omega(\epsilon\, v_{\max})$ on problem instance $\mathcal{I}_i$. Therefore, the (total) regret of $\mathcal{A}$ on problem instance $\mathcal{I}_i$ is $\Omega(\epsilon\, v_{\max} T) = \Omega(v_{\max}\, k^{1/3}\, T^{2/3})$. □

Theorem 3.6.3. *In the setting of Theorem 3.6.1, fix k and $v_{\max}$ and assume that $R(T;\, v_{\max}) = O(v_{\max}\, T^{\gamma})$ for some $\gamma < 1$. Then for every fixed $\delta \leq \frac{1}{4}$ and $\lambda < 2(1-\gamma)$ we have $R_\delta(T;\, v_{\max}) = \Omega(\delta\, v_{\max}\, T^{\lambda})$.*

Proof. Fix $\lambda \in (0,\, 2(1-\gamma))$. Redefine $\vec{\mu}_i$'s with respect to a different ϵ, namely $\epsilon = T^{-\lambda/2}$. Define the problem instances $\mathcal{I}_i$ in the same way as before: all bids are $v_{\max}$, the CTRs are given by $\vec{\mu}_i$.

Let us focus on agents 1 and 2. We claim that $\mathbb{E}_1[N_1] + \mathbb{E}_2[N_2] \geq \beta\, T^\lambda$, where $\beta > 0$ is a constant to be defined later. Suppose not. Fix all bids to be $v_{\max}$. For each round t, consider event $S_t = \{\mathcal{A}_t = 1\}$. Then by Claim 3.6.2 we have

$$\begin{aligned}
\left|\mathbb{P}_1[S_t] - \mathbb{P}_2[S_t]\right| &\leq \left|\mathbb{P}_0[S_t] - \mathbb{P}_1[S_t]\right| + \left|\mathbb{P}_0[S_t] - \mathbb{P}_2[S_t]\right| \\
&\leq O\left(\epsilon^2\right) \left(\mathbb{E}_1[N_1] + \mathbb{E}_2[N_2]\right) \\
&\leq \frac{1}{4},
\end{aligned}$$

for a sufficiently small β. Now, $\mathbb{P}_1[S_t] \geq \frac{1}{2}$ for at least $T/2$ rounds t. This is because otherwise on problem instance $\mathcal{I}_i$ regret would be $R(T) \geq \Omega(\epsilon\, T v_{\max}) = \Omega(v_{\max}\, T^{1-\lambda/2})$, which contradicts the assumption $R(T) = O(v_{\max}\, T^\gamma)$. Therefore $\mathbb{P}_2[S_t] \geq \frac{1}{4}$ for at least $T/2$ rounds t, hence on problem instance $\mathcal{I}_2$ regret is at least $\Omega(\epsilon\, T v_{\max})$, contradiction. Claim proved.

Now without loss of generality let us assume that $\mathbb{E}_1[N_1] \geq \frac{\beta}{2} T^\lambda$. Consider the problem instance in which CTRs given by $\vec{\mu}_1$, bid of agent 2 is $v_{\max}$, and all other bids are $v_{\max}(1-2\delta)/(1+2\epsilon)$. It is easy to see that this problem instance has δ-gap. Each time agent 1 is selected, algorithm incurs regret $\Omega(\delta v_{\max})$. Thus the total regret is at least $\Omega(\delta N_1\, v_{\max}) = \Omega(\delta\, v_{\max}\, T^\lambda)$. □

3.6.1 Relative entropy: Proof of Claim 3.6.2

In this section, we extend the relative entropy technique from Auer et al. (2002b). All relevant facts about relative entropy are summarized in the theorem below. We will need the following definition: given a random variable X on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, let $\mathbb{P}_X$ be the distribution of X, i.e. a measure on $\mathbb{R}$ defined by $\mathbb{P}_X(x) = \mathbb{P}[X = x]$.

Theorem 3.6.4. *Let p and q be two probability measures on a finite set U, and*

let Y and Z be functions on U. There exists a function $F(p;q|Y) : U \to \mathbb{R}$ with the following properties:

(i) $E_p\, F(p;q|Y) = E_p\, F(p;q|(Y,Z)) + E_p\, F(p_Z;q_Z|Y)$ *(chain rule),*

(ii) $\left|p(U') - q(U')\right| \leq \sqrt{\frac{1}{2}\mathcal{D}(p\|q)}$ *for any event* $U' \subset U$*, where* $\mathcal{D}(p\|q) = E_p\, F(p;q|1)$

(iii) for each $x \in U$, if conditional on the event $\{Z = Z(x)\}$ p coincides with q, then $F(p;q|Z)(x) = 0$.

(iv) for each $x \in U$, if conditional on the event $\{Z = Z(x)\}$ p and q are fair and $(\frac{1}{2}+\epsilon)$-biased coins, respectively, then it is the case that $F(p;q|Z)(x) \leq 4\epsilon^2$.

Remark. This theorem summarizes several well-known facts about relative entropy (albeit in a somewhat non-standard notation). For the proofs, see Cover and Thomas (1991); Kleinberg (2005, 2007b). In the proofs, one defines $F = F(p;q|Y)$ as a function $F : U \to \mathbb{R}$ which is specified by $F(x) = \sum_{x' \in U} p(x'|U_x) \lg \frac{p(x'|U_x)}{q(x'|U_x)}$, where U_x is the event $\{Y = Y(x)\}$.[10] Note that the quantity $E_p\, F(p;q|1)$ is precisely the relative entropy (a.k.a. KL-divergence), commonly denoted $\mathcal{D}(p\|q)$, and $E_p\, F(p;q|Y)$ is the corresponding conditional relative entropy.

In what follows we use Theorem 3.6.4 to prove Claim 3.6.2. For simplicity we will prove (3.6.1) for $i = 1$.

The *history* up to round t is $H_t = (h_1, h_2, \ldots, h_t)$ where $h_s \in \{0,1\}$ is the click or no click event received by the algorithm at round s. Let C_t be the indicator function of the event "round t is bid-independent". Define the *bid-independent history* as $\widehat{H}_t = (\widehat{h}_1, \widehat{h}_2, \ldots, \widehat{h}_t)$, where $\widehat{h}_t = h_t C_t$. For any exploration-separated deterministic allocation rule and each round t, the bid-independent history $\widehat{H}_{t-1}$ and the

[10] We use the convention that $p(x)\log(p(x)/q(x))$ is 0 when $p(x) = 0$, and $+\infty$ when $p(x) > 0$ and $q(x) = 0$.

bids completely determine which arm is chosen in this round. Moreover, $\widehat{H}_{t-1}$ alone (without the bids) completely determines whether round t is bid-independent, and if so, which arm is chosen in this round.

Recall the CTR vectors $\vec{\mu}_i$ as defined in Section 3.6. Let p and q be the distributions induced on $\widehat{H}_T$ by $\vec{\mu}_0$ and $\vec{\mu}_1$, respectively. Let p_t and q_t be the distributions induced on $\widehat{h}_t$ by $\vec{\mu}_0$ and $\vec{\mu}_1$, respectively. Let $\mathcal{H}_t$ the support of $\widehat{H}_t$, i.e. the set of all t-bit vectors. In the forthcoming applications of Theorem 3.6.4, the universe will be $U = \mathcal{H}_T$. By abuse of notation, we will treat $\widehat{H}_t$ as a projection $\mathcal{H}_T \to \mathcal{H}_t$, so that it can be considered a random variable under p or q.

Claim 3.6.5. $\mathcal{D}(p\|q) = E_p\, F(p; q|\, \widehat{H}_t) \;+\; \sum_{s=1}^{t} E_p\, F(p_s; q_s|\, \widehat{H}_{s-1})$ *for any* $t > 1$.

Proof. Use induction on $t \geq 0$ (set $\widehat{H}_0 = 1$). In order to obtain the claim for a given t assuming that it holds for $t-1$, apply Theorem 3.6.4(i) with $Y = \widehat{H}_{t-1}$ and $Z = \widehat{h}_t$. □

Claim 3.6.6. $F(p_t; q_t|\, \widehat{H}_{t-1}) \leq 4\epsilon^2\, C_t\, 1_{\{A_t=1\}}$ *for each round* t.

Proof. We are interested in the function $F = F(p_t; q_t|\, \widehat{H}_{t-1}) : \mathcal{H}_T \to \mathbb{R}$. Given $\widehat{H}_{t-1}$, one of the following three cases occurs:

- round t is not bid-independent. Then $\widehat{h}_t = 0$, hence $F(\cdot) = 0$ by Theorem 3.6.4(iii),
- round t is bid-independent and arm 1 is not played. Then $\widehat{h}_t$ is distributed as a fair coin under both p and q, so again $F(\cdot) = 0$.
- round t is bid-independent and arm 1 is played. Then $F(\cdot) \leq 4\epsilon^2$ by Theorem 3.6.4(iv). □

Given the full bid-independent history $\widehat{H}_T$, p and q become (the same) point measure, so by Theorem 3.6.4(iii) $E_p\, F(p;q|\,\widehat{H}_T) = 0$. Therefore taking Claim 3.6.5 with $t = T$ we obtain

$$\mathcal{D}(p\|q) = \sum_{t=1}^{T} E_p\, F(p_t;q_t|\,\widehat{H}_{t-1}) = 4\epsilon^2 \sum_{t=1}^{T} E_p\left[C_t\ 1_{\{A_t=1\}}\right] = 4\epsilon^2\ E_p[N_1]. \quad (3.6.2)$$

For a given round t and fixed bids, the allocation at round t is completely determined by the bid-independent history $\widehat{H}_{t-1}$. Thus, we can treat $\{A_t \in S\}$ as an event in $\mathcal{H}_T$. Now (3.6.1) follows from (3.6.2) via an application of Theorem 3.6.4(ii) with $U' = \{A_t \in S\}$.

3.7 Matching upper bound

Let us describe a very simple mechanism, called *the naive MAB mechanism*, which matches the lower bound from Theorem 3.6.1 up to polylogarithmic factors (and also the lower bound from Theorem 3.6.3, for $\gamma = \lambda = \frac{2}{3}$ and constant δ).[11]

Fix the number of agents k, the time horizon T, and the bid vector b. The mechanism has two phases. In the *exploration phase*, each agent is played for $T_0 := k^{-2/3}\, T^{2/3} (\log T)^{1/3}$ rounds, in a round robin fashion. Let c_i be the number of clicks on agent i in the exploration phase. In the *exploitation phase*, an agent $i^* \in \operatorname{argmax}_i c_i b_i$ is chosen and played in all remaining rounds. Payments are defined as follows: agent i^* pays $\max_{i\in[k]\setminus\{i^*\}} c_i b_i / c_{i^*}$ for every click she gets in exploitation phase, and all others pay 0. (Exploration rounds are free for every agent.) This completes the description of the mechanism.

[11] Independently, Devanur and Kakade Devanur and Kakade (2009) presented a version of the naive MAB mechanism that achieves the same regret even in the more general model in which the value-per-click of an agent changes over time and the agents are allowed to submit a different bid at every round. Instead of assigning all impressions to the same agent in the exploitation phase, their mechanism runs the same allocation and payment procedure for each exploration round separately (see Devanur and Kakade (2009) for details).

Observation 3.7.1. *Consider the stochastic MAB mechanism design problem with k agents. The naive mechanism is normalized, truthful and has worst-case regret $R(T; v_{\max}) = O(v_{\max}\, k^{1/3}\, T^{2/3}\, \log^{2/3} T)$.*

Proof. The mechanism is truthful by a simple second-price argument.[12] Recall that c_i is the number of clicks i got in the exploration phase. Let $p_i = \max_{j \neq i} c_j b_j / c_i$ be the price paid (per click) by agent i if she wins (all) rounds in exploitation phase. If $v_i \geq p_i$, then by bidding anything greater than p_i agent i gains $v_i - p_i$ utility each click irrespective of her bid, and bidding less than v_i, she gains 0, so bidding v_i is weakly dominant. Similarly, if $v_i < p_i$, then by bidding anything less than p_i she gains 0, while bidding $b_i > p_i$, she *loses* $b_i - p_i$ each click. So bidding v_i is weakly dominant in this case too.

For the regret bound, let $(\mu_1, \ \ldots \ , \mu_k)$ be the vector of CTRs, and let $\bar{\mu}_i = c_i / T_0$ be the sample CTRs. By Chernoff bounds, for each agent i we have $\Pr\left[|\bar{\mu}_i - \mu_i| > r\right] \leq T^{-4}$, for $r = \sqrt{8 \log(T)/T_0}$. If in a given run of the mechanism all estimates $\bar{\mu}_i$ lie in the intervals specified above, call the run *clean*. The expected regret from the runs that are not clean is at most $O(v_{\max})$, and can thus be ignored. From now on let us assume that the run is clean.

The regret in the exploration phase is at most

$$k\, T_0\, v_{\max} = O\left(v_{\max}\, k^{1/3}\, T^{2/3}\, \log^{1/3} T\right).$$

For the exploitation phase, let $j = \operatorname{argmax}_i \mu_i b_i$. Then (since we assume that the run is clean) we have

$$(\mu_{i^*} + r)\, b_{i^*} \geq \bar{\mu}_{i^*}\, b_{i^*} \geq \bar{\mu}_j\, b_j \geq (\mu_j - r)\, b_j,$$

[12] Alternatively, one can use Theorem 3.5.8 since all exploration rounds are bid-independent, and only exploration rounds are influential, and the payments are exactly as defined in Theorem 3.5.1.

which implies $\mu_j v_j - \mu_{i^*} v_{i^*} \leq r(v_j + v_{i^*}) \leq 2r\, v_{\max}$. Therefore, the regret in exploitation phase is at most $2r\, v_{\max}\, T = O(v_{\max}\, k^{1/3}\, T^{2/3}\, \log^{2/3} T)$. Therefore the total regret is as claimed. □

3.8 Extensions

We extend our results in several directions. In this section, we first describe the extensions, and then present them in turn in subsequent subsections.

First, we derive a regret lower bound for deterministic truthful mechanisms without assuming that the allocations are scale-free. In particular, for two agents there are no assumptions. This lower bound holds for any k (the number of agents) assuming IIA, but unlike the one in Theorem 3.6.1 it does not depend on k.

Second, we extend our results to randomized mechanisms. We consider randomized mechanisms that are *universally truthful*, i.e. truthful for each realization of the internal random seed. For mechanisms that randomize over exploration-separated deterministic allocation rules, we obtain the same lower bounds as in Theorems 3.6.1 and Theorem 3.6.3.

Third, we consider randomized allocation rules under a weaker version of truthfulness: a mechanism is *weakly truthful* if for each realization, it is truthful in expectation over its random seed. We show that any randomized allocation that is "pointwise monotone" and satisfies a certain notion of "separation between exploration and exploitation" can be turned into a mechanism that is weakly truthful and normalized. Then we apply this result to an algorithm in the literature (Awerbuch and Kleinberg, 2008; Kleinberg, 2007a) in order to obtain regret guarantees for the version of the MAB mechanism design problem in which the clicks are

chosen by an oblivious adversary.[13] (This version corresponds to the *adversarial MAB problem* (Auer et al., 2002b; Dani and Hayes, 2006; Abernethy et al., 2008; Bartlett et al., 2008).) The upper bound matches our lower bound for deterministic allocations up to $(\log k)^{1/3}$ factors.

Fourth, we consider the stochastic MAB mechanism design problem under a more relaxed notion of truthfulness: truthfulness *in expectation*, where for each vector of CTRs the expectation is taken over clicks (and the internal randomness in the mechanism, if the latter is not deterministic). Following our line of investigation, we ask whether restricting a mechanism to be truthful in expectation has any implications on the structure and regret thereof. Given our results on mechanisms that are truthful and normalized, it is tempting to seek similar results for mechanisms that are truthful in expectation and normalized in expectation.[14] We rule out this approach: we show that in order to obtain any non-trivial lower bounds on regret and (essentially) any non-trivial structural results, one needs to assume that a mechanism is ex-post normalized, at least in some approximate sense. The key idea is to view the allocation and the payment as multivariate polynomials over the CTRs.

We now describe these extensions in turn.

3.8.1 Lower bound for non-scalefree allocations

In this section we derive a regret lower bound for deterministic truthful mechanisms without assuming that the allocations are scale-free. In particular, for two agents

[13] An oblivious adversary chooses the entire realization in advance, without observing algorithm's behavior and its random seed.

[14] A mechanism is *normalized in expectation* if in expectation over clicks (and possibly over the allocation's randomness), each agent is charged an amount between 0 and her bid for each click she receives.

there are no assumptions. This lower bound holds for any k (the number of agents) assuming that the allocation satisfies IIA, but unlike the one in Theorem 3.6.1 it does not depend on k.

Theorem 3.8.1. *Consider the stochastic MAB mechanism design problem with k agents. Let $(\mathcal{A}, \mathcal{P})$ be a normalized truthful mechanism such that $\mathcal{A}$ is a non-degenerate deterministic allocation rule. Suppose $\mathcal{A}$ satisfies IIA. Then its regret is $R(T; v_{\max}) = \Omega(v_{\max} T^{2/3})$ for any sufficiently large $v_{\max}$.*

Note that the theorem does not assume scalefreeness of the mechanism.

Let us sketch the proof. Fix an allocation $\mathcal{A}$. In Definition 3.5.4, if round t is (b, ρ) influential, for some realization ρ and bid vector b, an agent i is called *strongly influenced* by round t if it is one of the two agents that are "influenced" by round t but is not the "influencing agent" of round t. In particular, it holds that $\mathcal{A}(b, \rho, t) \neq i$. For each realization ρ, round t and agent i, if there exists a bid vector b such that round t is (b, ρ)-influential with strongly influenced agent i, then fix any one such b, and define $b_i^* = b_i^*(\rho, t) := \max_{j \neq i} b_j$. Let us define $B^*_{\mathcal{A}} = \max_{\rho,t,i} b_i^*(\rho, t)$, where the maximum is taken over all realizations ρ, all rounds t, and all agents i. Let us say that round t is *B^*-free* from agent i w.r.t realization ρ, if for this realization the following property holds: agent i is not selected in round t as long as each bid is at least B^*.

Lemma 3.8.2. *In the setting of Theorem 3.8.1, for any realization ρ, any influential round t is $B^*_{\mathcal{A}}$-free from some agent w.r.t. ρ.*

Proof. Fix realization ρ. Since round t is influential, for some bid profile b and agent i it is (b, ρ)-influential with a strongly influenced agent i. By definition of

$b_i^*(\rho, t)$, without loss of generality each bid in b (other than i's bid) is at most $b_i^*(\rho, t) \leq B_{\mathcal{A}}^*$. Then $\mathcal{A}(b, \rho, t) \neq i$, and round t is (b, ρ)-secured from agent i.

Suppose round t is not $B_{\mathcal{A}}^*$-free from agent i w.r.t ρ. Then there exists a bid profile b' in which each bid (other than i's bid) is at least $B_{\mathcal{A}}^*$ such that $\mathcal{A}(b', \rho, t) = i$. To derive a contradiction, let us transform b to b' by adjusting first the bid of agent i and then bids of agents $j \neq i$ one agent at a time. Initially agent i is not chosen in round t, and after the last step of this transformation agent i is chosen. Thus it is chosen at some step, say when we adjust the bid of agent i or some agent $j \neq i$. This *transfer of impression* to agent i cannot happen when bid of agent i is adjusted from b_i to b_i' (since round t is $(b; \rho)$-secured from i), and it cannot happen when bid of player $j \neq i$ is adjusted from b_j to $b_j' \geq b_j$ (this is because, the transfer to i cannot happen from j because of pointwise-monotonicity and the transfer to i cannot happen from $l \neq j$ because of IIA). This is a contradiction. □

Let T be the time horizon. Assume $v_{\max} \geq 2B_{\mathcal{A}}^*$. Let $N(\rho)$ be the number of influential rounds w.r.t realization ρ. Let $N_i(\rho)$ be the number of influential rounds w.r.t. realization ρ that are $B_{\mathcal{A}}^*$-free from agent i w.r.t. ρ. Then N and the N_i's are random variables in the probability space induced by the clicks. By Lemma 3.8.2 we have that $\sum_i N_i(\rho)$ is at least the number of *influential rounds*. As in Section 3.6, let $\vec{\mu}_0$ be the vector of CTRs in which all CTRs are $\frac{1}{2}$, and let $\mathbb{E}_0[\cdot]$ denote expectation w.r.t. $\vec{\mu}_0$.

Fix a constant $\beta > 0$ to be specified later. If $\mathbb{E}_0[N] \geq \beta k T^{2/3}$ then $\mathbb{E}_0[N_i] \geq \beta T^{2/3}$ for some agent i, so the allocation incurs expected regret $R(T; v_{\max}) \geq \Omega(v_{\max} T^{2/3})$ on any problem instance $\mathcal{J}_j$, $j \neq i$. (In this problem instance, CTRs given by $\vec{\mu}_0$, the bid of agent j is $v_{\max}$, and all other bids are $v_{\max}/2$.) Now suppose $\mathbb{E}_0[N] \leq \beta k T^{2/3}$. Then the desired regret bound follows by an argument

very similar to the one in the last paragraph of the proof of Theorem 3.6.1.

3.8.2 Universally truthful randomized mechanisms

Consider randomized mechanisms that are *universally truthful*, i.e. truthful for each realization of the internal random seed. For mechanisms that randomize over exploration-separated deterministic mechanisms, we obtain the same lower bounds as in Theorems 3.6.1 and Theorem 3.6.3.

Theorem 3.8.3. *Consider the MAB mechanism design problem. Let $\mathcal{D}$ distribution over exploration-separated deterministic allocation rules. Then*

$$\mathbb{E}_{\mathcal{A}\in\mathcal{D}}\left[R_{\mathcal{A}}(T; v_{\max})\right] = \Omega(v_{\max}\, k^{1/3}\, T^{2/3}).$$

Proof Sketch. Recall that in the proof of Theorem 3.6.1 we define a family $\mathcal{F}$ of $2k$ problem instances, and show that if $\mathcal{A}$ is an exploration-separated deterministic allocation rule, then on one of these instances its regret is "high". In fact, we can extend this analysis to show that the regret is "high", that is at least $R^* = \Omega(v_{\max}\, k^{1/3}\, T^{2/3})$, on an instance $\mathcal{I} \in \mathcal{F}$ chosen uniformly at random from $\mathcal{F}$; here regret is in expectation over the choice of $\mathcal{I}$.[15] Once this is proved, it follows that regret is $R^*/2$ for any *distribution* over such $\mathcal{A}$, in expectation over both the choice of $\mathcal{A}$ and the choice of $\mathcal{I}$. Thus there exists a single (deterministic) instance $\mathcal{I}$ such that $\mathbb{E}_{\mathcal{A}\in\mathcal{D}}\left[R_{\mathcal{A},\mathcal{I}}(T)\right] \geq R^*/2$. □

Theorem 3.6.3 extends similarly.

[15] This extension requires but minor modifications to the proof of Theorem 3.6.1. For instance, for the case $k \geq 3$ we argue that first, if $\mathbb{E}_0[N] > R$ then $\mathbb{E}_0[N_i] \leq \frac{2}{k} E_0[N]$ for at least $\frac{k}{2}$ agents i (and so on), and if $\mathbb{E}_0[N] \leq R$ then (omitting some details) there are $\Omega(k)$ good agents i such that $\mathbb{E}_0[N_i] \leq 2R/k$ (and so on).

3.8.3 Randomized allocations and adversarial clicks

In this section we discuss randomized allocations and the version of the MAB mechanism design problem when clicks are generated adversarially, termed the *adversarial MAB problem*. In this version, the objective is to optimize the worst-case regret over all values $v = (v_1, \ldots, v_k)$ such that $v_i \in [0, v_{\max}]$ for each i, *and all realizations* ρ:

$$R(T; v; \rho) = \left[\max_i v_i \sum_{t=1}^{T} \rho_i(t) \right] - \sum_{t=1}^{T} \sum_{i=1}^{k} v_i \, \rho_i(t) \, \mathbb{E}\left[\mathcal{A}_i(v; \rho; t)\right] \tag{3.8.1}$$

$$R(T; v_{\max}) = \max\{R(T; v; \rho) : \text{ all realizations } \rho,$$
$$\text{all } v \text{ such that } v_i \in [0, v_{\max}] \text{ for each } i\}.$$

The first term in (3.8.1) is the social welfare from the best time-invariant allocation, the second term is the social welfare generated by $\mathcal{A}$.

Let us make a few definitions related to truthfulness. Recall that a mechanism is called *weakly truthful* if for each realization, it is truthful in expectation over its random seed. A randomized allocation is *pointwise monotone* if for each realization and each bid profile, increasing the bid of any one agent does not decrease the probability of this agent being allocated in any given round. For a set S of rounds and a function $\sigma : S \to \{\text{agents}\}$, an allocation is (S, σ)-*separated* if (i) it coincides with σ on S, (ii) the clicks from the rounds not in S are discarded (not reported to the algorithm). An allocation is *strongly separated* if before round 1, without looking at the bids, it randomly chooses a set S of rounds and a function $\sigma : S \to \{\text{agents}\}$, and then runs a pointwise monotone (S, σ)-separated allocation. Note that the choice of S and σ is independent of the clicks, by definition.

We show that for any (randomized) strongly separated allocation rule $\mathcal{A}$ there exists a payment rule which results in a mechanism that is weakly truthful and

normalized. Then we consider PSim (Awerbuch and Kleinberg, 2008; Kleinberg, 2007a), a randomized MAB algorithm from the literature, and show that it is pointwise monotone and strongly separated. When interpreted as an allocation rule, there algorithm has strong regret guarantees for the *adversarial MAB mechanism design problem*, where the clicks are chosen by an *oblivious adversary*. Specifically, PSim obtains regret $R(T, v_{\max}) = O(v_{\max}\, k^{1/3}\, (\log k)^{1/3}\, T^{2/3})$.

We start with the structural result.

Lemma 3.8.4. *Consider the MAB mechanism design problem. Let $\mathcal{A}$ be a (randomized) strongly separated allocation rule. Then there exists a payment rule $\mathcal{P}$ such that the resulting mechanism $(\mathcal{A}, \mathcal{P})$ is normalized and weakly truthful.*

Proof. Throughout the proof, let us fix a realization ρ, time horizon T, bid vector b, and agent i. We will consider the payment of agent i. We will vary the bid of agent i on the interval $[0, b_i]$; the bids b_{-i} of all other agents always stay the same.

Let $c_i(x)$ be the number of clicks received by agent i given that her bid is x. Then by (the appropriate version of) Theorem 3.5.1 the payment of agent i must be $\mathcal{P}_i(b)$ such that

$$\mathbb{E}_{\mathcal{A}}[\mathcal{P}_i(b)] = \mathbb{E}_{\mathcal{A}}\left[b_i\, c_i(b_i) - \textstyle\int_{x=0}^{b_i} c_i(x)\, dx\right], \tag{3.8.2}$$

where the expectation is taken over the internal randomness in the algorithm.

Recall that initially $\mathcal{A}$ randomly selects, without looking at the bids, a set S of rounds and a function $\sigma : S \to \{\text{agents}\}$, and then runs some pointwise monotone (S, σ)-separated allocation $\mathcal{A}^{(S,\sigma)}$. In what follows, let us fix S and σ, and denote $\mathcal{A}^* = \mathcal{A}^{(S,\sigma)}$. We will refer to the rounds in S as *exploration rounds*, and to the rounds not in S as *exploitation rounds*. Let $\gamma_i^*(x, t)$ be the probability

that algorithm $\mathcal{A}^*$ allocates agent i in round t given that agent i bids x. Note that for fixed value of internal random seed of $\mathcal{A}^*$ this probability can only depend on the clicks observed in exploration rounds, which are known to the mechanism. Therefore, abstracting away the computational issues, we can assume that it is known to the mechanism. Define the payment rule as follows: in each exploitation round t in which agent i is chosen and clicked, charge

$$\mathcal{P}_i^*(b,t) = b_i - \frac{1}{\gamma_i^*(b_i,t)} \int_0^{b_i} \gamma_i^*(x,t)\,dx. \tag{3.8.3}$$

Then the total payment assigned to agent i is

$$\mathcal{P}_i^*(b) = \sum_{t \notin S} \rho_i(t)\ \mathcal{A}_i^*(b;\rho;t)\ \mathcal{P}_i^*(b,t). \tag{3.8.4}$$

Since allocation $\mathcal{A}^*$ is pointwise monotone, the probability $\gamma_i^*(x,t)$ is non-decreasing in x. Therefore $\mathcal{P}_i^*(b,t) \in [0,b_i]$ for each round t. It follows that the mechanism is normalized (for any realization of the random seed of allocation $\mathcal{A}$).

It remains to check that the payment rule (3.8.3) results in (3.8.2). Let $c_i^*(x)$ be the number of clicks allocated to agent i by allocation $\mathcal{A}^*$ given that her bid is x. Let $c_i^{\text{expl}}(x)$ be the corresponding number of clicks in exploitation rounds only. Since $\mathcal{A}^*$ is (S,σ)-separated, we have

$$\mathbb{E}[c_i^*(x) - c_i^{\text{expl}}(x)] = \sum_{t \in S} \rho_{\sigma(t)}(t)\ = \text{const}(x). \tag{3.8.5}$$

Taking expectations in (3.8.4) over the random seed of $\mathcal{A}_S$ and using (3.8.5), we obtain

$$\begin{aligned}
\mathbb{E}[\mathcal{P}_i^*(b)] &= \sum_{t \notin S} \rho_i(t)\ \gamma_i^*(b_i,t)\ \mathcal{P}_i^*(b,t) \\
&= \sum_{t \notin S} \rho_i(t)\ \left[b_i\,\gamma_i^*(b_i,t) - \int_0^{b_i} \gamma_i^*(x,t)\,dx \right]
\end{aligned}$$

$$= b_i \left[\sum_{t \notin S} \rho_i(t)\, \gamma_i^*(b_i, t) \right] - \int_0^{b_i} \left[\sum_{t \notin S} \rho_i(t)\, \gamma_i^*(x, t) \right] dx$$
$$= b_i\, \mathbb{E}\left[c_i^{\text{expl}}(b_i)\right] - \int_0^{b_i} \mathbb{E}[c_i^{\text{expl}}(x)]\, dx$$
$$= \mathbb{E}\left[b_i\, c_i^*(b_i) - \int_0^{b_i} c_i^*(x)\, dx \right].$$

Finally, taking expectations over the choice of S and σ, we obtain (3.8.2). □

Algorithm PSIM is strongly separated

In this subsection, we consider PSIM (Awerbuch and Kleinberg, 2008; Kleinberg, 2007a), an algorithm for the adversarial MAB problem. We interpret this algorithm as an allocation rule, and observe that it is strongly separated.

As usual, k denotes the number of agents; let $[k]$ denote the set of agents. The algorithm is shown in Figure 3.2.

If we pick the values $\epsilon = (k \log k/T)^{1/3}$ and $P = (\log k)^{1/3}(T/k)^{2/3}$, then the regret of PSIM is bounded by $\mathcal{O}((k \log k)^{1/3} T^{2/3} v_{\max})$ against any oblivious adversary (see (Awerbuch and Kleinberg, 2008; Kleinberg, 2007a)).

We next prove that PSIM is strongly-separated.

It is clear from the structure of PSIM above that it chooses a set S of exploration rounds and a function $f : S \to [k]$ in the beginning without looking at the bids and then runs an (S, f)-separated allocation. We need to prove that the (S, f)-separated allocation is pointwise monotone. For this we need prove that the probability $\gamma_i(b; t; S, f)$ is monotone in the bid of agent i, where $\gamma_i(b; t; S, f)$ denotes the probability of picking agent i in round t when bids are b given the choice of S and f. If $t \in S$, the $\gamma_i(b; t; S, f)$ is independent of bids, and hence is

INPUT: Time horizon T, bid vector b. Let $v_{\max} = \max_i b_i$.

OUTPUT: For each round $t \leq T$, a distribution on $[k]$.

Divide the time horizon into P phases of T/P consecutive rounds each.

From rounds of each phase p, pick without replacement some k rounds at random (called the exploration rounds) and assign them randomly to k arms. Let S denote the set of all exploration rounds (of all phases). Let $f : S \to [k]$ be the function which tells which arm is assigned to an exploration round in S. The rounds in $[T] \setminus S$ are called the exploitation rounds.

Let $w_i(0) = 1$ for all $i \in [k]$.

FOR each phase $p = 1, 2, \ldots, P$

    FOR each round t in phase p

        IF $t \in S$ and $f(t) = i$

            Define the distribution $\gamma(b; t; S, f)$ such that $\gamma_i(b; t; S, f) = 1$.

            Pick an agent according to this distribution (equivalently, pick agent i), observe the click $\rho_i(t)$, and update $w_i(p)$ multiplicatively by $w_i(p) = w_i(p-1) \cdot (1+\epsilon)^{\rho_i(t) b_i / v_{\max}}$.

        IF $t \notin S$

            Define the distribution $\gamma(b; t; S, f)$ such that $\gamma_i(b; t; S, f) = \frac{w_i(p-1)}{\sum_j w_j(p-1)}$.

            Pick an agent according to $\gamma(b; t; S, f)$, observe the feedback, and discard the feedback.

Figure 3.2: The PSim algorithm.

monotone in b_i. Let $t \notin S$ and t is a round in phase p. Let us denote by $f^{-1}(i,p)$ the (unique) exploration round in phase p assigned to agent i. We then have

$$\gamma_i(b;t;S,f) = (1+\epsilon)^{\frac{b_i}{v_{\max}} \sum_{q=1}^{p-1} \rho_i(f^{-1}(i,q))} \Big/ \sum_j (1+\epsilon)^{\frac{b_j}{v_{\max}} \sum_{q=1}^{p-1} \rho_j(f^{-1}(j,q))}.$$

We split the denominator into the term for agent i and all other terms. It is then not hard to see that this is a non-decreasing function of b_i.

We state the above results in the form of the following corollary.

Corollary 3.8.5. *There exists a weakly truthful normalized mechanism for the adversarial MAB problem (against oblivious adversary) whose regret grows as* $\mathcal{O}((k \log k)^{1/3} \cdot T^{2/3} \cdot v_{\max})$.

3.8.4 Truthfulness in expectation over CTRs

We consider the stochastic MAB mechanism design problem under a more relaxed notion of truthfulness: truthfulness *in expectation*, where for each vector of CTRs the expectation is taken over clicks (and the internal randomness in the mechanism, if the latter is not deterministic). We show that any allocation $\mathcal{A}^*$ that is monotone in expectation,[16] can be converted to a mechanism that is truthful in expectation and monotone in expectation, with minor changes and a very minor increase in regret. Furthermore, we show that there exist MAB allocations that are monotone in expectation whose regret matches the optimal upper bounds for MAB *algorithms*. The conclusion is that in order to obtain any non-trivial lower bounds on regret and (essentially) any non-trivial structural results, one needs

[16]Monotonicity in expectation is defined in an obvious way: an allocation is *monotone in expectation* if for each agent i and fixed bid profile b_{-i}, the corresponding expected click-allocation is a non-decreasing function of b_i; here the expectation is taken over the clicks and possibly the allocation's random seed.

to assume that a mechanism is ex-post normalized, at least in some approximate sense.

The main result of this section is that for any allocation $\mathcal{A}^*$ that is monotone in expectation, any time horizon T, and any parameter $\gamma \in (0,1)$ there exists a mechanism $(\mathcal{A}, \mathcal{P})$ such that the mechanism is truthful in expectation and normalized in expectation, and allocation $\mathcal{A}$ initially makes a random choice between $\mathcal{A}^*$ and some other allocation, choosing $\mathcal{A}^*$ with probability at least γ. We call such allocation $\mathcal{A}$ a γ-*approximation* of $\mathcal{A}^*$. Clearly, on any problem instance we have $R_{\mathcal{A}}(T) \leq \gamma\, R_{\mathcal{A}^*}(T) + (1-\gamma)T$. The extra additive factor of $(1-\gamma)T$ is not significant if e.g. $\gamma = 1 - \frac{1}{T}$. The problem with this mechanism is that it is not ex-post normalized; moreover, in some realizations payments may be very large in absolute value.

Theorem 3.8.6. *Consider the stochastic MAB mechanism design problem with k agents and a fixed time horizon T. For each $\gamma \in (0,1)$ and each allocation rule $\mathcal{A}^*$ that is monotone in expectation, there exists a mechanism $(\mathcal{A}, \mathcal{P})$ such that $\mathcal{A}$ is a γ-approximation of $\mathcal{A}^*$, and the mechanism is truthful in expectation and normalized in expectation.*

Remark. Payment rule $\mathcal{P}$ is well-defined as a mapping from histories to numbers. We do not make any claims on the efficient computability thereof.

For the sake of completeness, we provide a concrete algorithm which one could plug into Theorem 3.8.6 and obtain improved (and in fact, best possible) regret guarantees.

Proposition 3.8.7. *Consider the stochastic MAB mechanism design problem with k agents and a fixed time horizon T. There exists an allocation rule $\mathcal{A}$ that is*

monotone in expectation, whose regret is $R(T;\, v_{\max}) = O(v_{\max}\sqrt{kT\log T})$ *in the worst case, and* $R_\delta(T;\, v_{\max}) = O(v_{\max}\frac{k}{\delta}\log T)$ *on the* δ*-gap instances.*

Proof Sketch. For simplicity, assume $v_{\max} = 1$. Let $r_0 = \sqrt{8\log(T)/T}$. Consider the following simple allocation. Initially, each agent is *active*. In each phase, play each active agent once, in a round-robin fashion. After the phase, (permanently) de-activate each agent whose *sample product* (sample average times the bid) is more than r_0 below that of some other active agent. This completes the description of the allocation.

This allocation is based on a well-known (perhaps folklore) MAB algorithm. The regret bounds are proved along the lines of those in Auer et al. (2002b) (also see Section 2.3). The crucial observations are that with a very high probability the optimal agent is never de-activated, and that that each sub-optimal agent i is played at most $O(\Delta_i^{-2}\log T)$ times, where Δ_i is the difference between her product (CTR times the bid) and the maximal one.

The allocation is monotone in expectation because increasing the bid of a given agent cannot cause this agent to be de-activated later. □

Proof of Theorem 3.8.6

Let $\mathcal{A}_{\text{expl}}$ be the allocation rule where in each round an agent is chosen independently and uniformly at random. Allocation $\mathcal{A}$ is defined as follows: use $\mathcal{A}^*$ with probability γ; otherwise use $\mathcal{A}_{\text{expl}}$. Fix an instance (b, μ) of the stochastic MAB mechanism design problem, where $b = (b_1, \ldots, b_k)$ and $\mu = (\mu_1, \ldots, \mu_k)$ are vectors of bids and CTRs, respectively. Let $C_i = C_i(b_i; b_{-i})$ be the expected number of clicks for agent i under the original allocation $\mathcal{A}^*$. Then by Myerson (1981)

the expected payment of agent i must be

$$\mathcal{P}_i^{\mathrm{M}} = \gamma \left[b_i \, C_i(b_i; b_{-i}) - \int_0^{b_i} C_i(x; b_{-i}) \, dx \right]. \tag{3.8.6}$$

The key idea is to treat the expected payment as a multivariate polynomial over $\mu_1, \ldots, \mu_k$. It is essential (given the way we define $\mathcal{P}$) to show that this polynomial has degree $\leq T$.

Claim 3.8.8. *$\mathcal{P}_i^M$ is a polynomial of degree $\leq T$ in variables $\mu_1, \ldots, \mu_k$.*

Proof. Fix the bid profile. Let X_t be allocation of algorithm $\mathcal{A}^*$. Let $\texttt{poly}(T)$ be the set of all polynomials over $\mu_1, \ldots, \mu_k$ of degree at most T. Consider a fixed history $h = (x_1, y_1; \ \ldots \ ; x_T, y_T)$, and let h^t be the corresponding history up to (and including) round t. Then

$$\mathbb{P}[h] = \textstyle\prod_{t=1}^{T} \Pr[X_t = x_t \,|\, h^{t-1}] \ \ \mu_{x_t}^{y_t} \, (1 - \mu_{x_t})^{1-y_t} \in \texttt{poly}(T) \tag{3.8.7}$$

$$C_i(b_i; b_{-i}) = \textstyle\sum_{h \in \mathcal{H}} \mathbb{P}[h] \ \#\mathrm{clicks}_i(h) \ \in \texttt{poly}(T). \tag{3.8.8}$$

Therefore $\mathcal{P}_i^{\mathrm{M}} \in \texttt{poly}(T)$, since one can take an integral in (3.8.6) separately over the coefficient of each monomial of $C_i(x; b_{-i})$. □

Fix time horizon T. For a given run of an allocation rule, the *history* is defined as $h = (x_1, y_1; \ \ldots \ ; x_T, y_T)$, where x_t is the allocation in round t, and $y_t \in \{0, 1\}$ is the corresponding click. Let $\mathcal{H}$ be the set of all possible histories.

Our payment rule $\mathcal{P}$ is a deterministic function of history. For each agent i, we define the payment $\mathcal{P}_i = \mathcal{P}_i(h)$ for each history h such that $E_h[\mathcal{P}_i(h)] = \mathcal{P}_i^{\mathrm{M}}$ for any choice of CTRs, and hence $E_h[\mathcal{P}_i(h)] \equiv \mathcal{P}_i^{\mathrm{M}}$, where $\equiv$ denotes an equality between polynomials over $\mu_1, \ldots, \mu_k$.

Fix the bid vector and fix agent i. We define the payment $\mathcal{P}_i$ as follows. Charge nothing if allocation $\mathcal{A}^*$ is used. If allocation $\mathcal{A}_{\text{expl}}$ is used, charge *per monomial*. Specifically, let $\texttt{mono}(T)$ be the set of all monomials over $\mu_1, \ldots, \mu_k$ of degree at most T. For each monomial $Q \in \texttt{mono}(T)$ we define a subset of *relevant histories* $\mathcal{H}_i(Q) \subset \mathcal{H}$. (We defer the definition till later in the proof.) For a given history $h \in \mathcal{H}$ we charge a (possibly negative) amount

$$\mathcal{P}_i(h) = \tfrac{1}{1-\gamma} \sum_{Q \in \texttt{mono}(T):\, h \in \mathcal{H}_i(Q)} k^{\deg(Q)}\, \mathcal{P}_i^{\text{M}}(Q), \tag{3.8.9}$$

where $\deg(Q)$ is the degree of Q, and $\mathcal{P}_i^{\text{M}}(Q)$ is the coefficient of Q in $\mathcal{P}_i^{\text{M}}$. Let $\mathbb{P}_{\text{expl}}$ be the distribution on histories induced by $\mathcal{A}_{\text{expl}}$. Then the expected payment is

$$E_h[\mathcal{P}_i(h)] = \sum_{Q \in \texttt{mono}(T)} k^{\deg(Q)}\, \mathbb{P}_{\text{expl}}[\mathcal{H}_i(Q)]\, \mathcal{P}_i^{\text{M}}(Q).$$

Therefore in order to guarantee that $E_h[\mathcal{P}_i(h)] \equiv \mathcal{P}_i^{\text{M}}$ it suffices to choose $\mathcal{H}_i(Q)$ for each Q so that

$$k^{\deg(Q)}\, \mathbb{P}_{\text{expl}}[\mathcal{H}_i(Q)] \equiv Q. \tag{3.8.10}$$

Consider a monomial $Q = \mu_1^{\alpha_1} \ldots \mu_k^{\alpha_k}$. Let $\mathcal{H}_i(Q)$ consist of all histories such that first agent 1 is played α_1 times in a row, and clicked every time, then agent 2 is played α_2 times in a row, and clicked every time, and so on till agent k. In the remaining $T - \deg(Q)$ rounds, any agent can be chosen, and any outcome (click or no click) can be received. It is clear that (3.8.10) holds.

CHAPTER 4

SLEEPING EXPERTS AND BANDITS PROBLEM

In this chapter, we describe a version of the online-learning problem where the set of available actions is allowed to vary over time. We repeat the common definition from the introduction, to remind the reader of the settings of this problem.

4.1 Introduction

As described in Chapter 1, in on-line decision problems, or sequential prediction problems, an algorithm must choose, in each of the T consecutive rounds, one of the n possible actions (as a slight change of notation from previous chapters, we use n to denote the number of options, instead of K). In each round, each action receives a real valued positive payoff in $[0, 1]$, initially unknown to the algorithm. At the end of each round the algorithm receives some information about the payoffs of the actions in that round. The goal of the algorithm is to maximize the total payoff, i.e. the sum of the payoffs of the chosen actions in each round. The standard on-line decision settings are the *best expert* setting (or the full-information setting) in which, at the end of the round, the payoffs of *all* n strategies are revealed to the algorithm, and the *multi-armed bandit* setting (or the partial-information setting) in which only the payoff of the chosen strategy is revealed. Customarily, in the best expert setting the strategies are called *experts* and in the multi-armed bandit setting the strategies are called *bandits* or *arms*. We use *actions* to generically refer to both types of strategies, when we do not refer particularly to either.

In the prior-free setting (as is the case in this chapter), the performance of the algorithm is typically measured in terms of *regret*. (See (Gittins, 1979), (Gittins

and Jones, 1979) for maximization of expected reward in the Bayesian setting.) The regret is the difference between the expected payoff of the algorithm and the payoff of a single fixed strategy for selecting actions. The usual single fixed strategy to compare against is the one which always selects the expert or bandit that has the highest total payoff over the T rounds in hindsight.

The usual assumption in online learning problems is that all actions are available at all times. In many applications, however, this assumption is not appropriate. In network routing problems, for example, some of the routes are unavailable at some point in time due to router or link crashes. Or, in electronic commerce problems, items are out of stock, sellers are not available (due to maintenance or simply going out of business), and buyers do not buy all the time. Even in the setting that gave multi-armed bandit problems their name, a gambler playing slot machines, some of the slot machines might be occupied by other players at any given time.

In this chapter we relax the assumption that all actions are available at all times, and allow the set of available actions to vary in an *adversarial* way from one round to the next, a model known as "predictors that specialize" or "sleeping experts" in prior work. The first foundational question that needs to be addressed is how to define regret when the set of available actions may vary over time. Defining regret with respect to the best action in hindsight is no longer appropriate since that action might sometimes be unavailable. A useful thought experiment for guiding our intuition is the following: if each action had a fixed payoff distribution that was *known* to the decision-maker, what would be the best way to choose among the available actions? The answer is obvious: one should order all of the actions according to their expected payoff, then choose among the available actions by

selecting the one which ranks highest in this ordering. Guided by the outcome of this thought experiment, we define our base to be the best ordering of actions in hindsight (see Section 4.1.1 for a formal definition) and contend that this is a natural and intuitive way to define regret in our setting. This contention is also supported by the informal observation that order-based decision rules seem to resemble the way people make choices in situations with a varying set of actions, e.g. choosing which brand of beer to buy at a store.

We prove lower and upper bounds on the regret with respect to the best ordering for both the best expert setting and the multi-armed bandit setting. We first explore the case of a stochastic adversary, where the payoffs received by action i at each time step are independent samples from an unknown but fixed distribution $P_i(\cdot)$ supported on $[0, 1]$ with mean μ_i. (Note that in this work, the choice of which actions are available to be picked in each round is always adversarial. In other words, there is no distributional assumption on the subset of available actions.) Assuming that $\mu_1 > \mu_2 > \cdots > \mu_n$ (and the algorithm, of course, does not know the identities of these actions) we show that the regret of any learning algorithm will necessarily be at least $\Omega\left(\sum_{i=1}^{n-1} \frac{1}{\mu_i - \mu_{i+1}}\right)$ in the best expert setting, and $\Omega\left(\log(T) \sum_{i=1}^{n-1} \frac{1}{\mu_i - \mu_{i+1}}\right)$ in the multi-armed bandit setting if the game is played for T rounds (for T sufficiently large[1]). We also present efficient learning algorithms for both settings. For the multi-armed bandit setting, our algorithm, called AUER, is an adaptation of the UCB1 algorithm in Auer et al. (2002a), which comes within a constant factor of the lower bound mentioned above. For the expert setting, a very simple algorithm, called "follow-the-awake-leader", which is a variant of "follow-the-leader" (Hannan, 1957; Kalai and Vempala, 2005), comes within a

[1] As is the convention in the literature, the problem instance is not allowed to depend on T in the stochastic setting. In other words, first the problem instance is chosen, and then we look at regret bounds as a function of T.

constant factor of the lower bound above. While our algorithms are adaptations of existing techniques, the proofs of the upper and lower bounds hinge on some technical innovations.

For the lower bound in stochastic multi-armed bandit setting, we must modify the classic asymptotic lower bound proof of Lai and Robbins (1985b) to obtain a bound which holds at all sufficiently large finite times. For the stochastic best expert setting, we adapt standard KL-divergence arguments to prove a precise lower bound that also holds for sufficiently large finite times. Our lower bounds in Lemma 4.2.8 and Lemma 4.2.14 don't refer to the "sleeping" version of the problem, and concern the classical best-expert setting and multi-armed bandit setting (all actions available), which might be of interest outside the context of this work.

To prove that our lower and upper bounds are within a constant factor of each other we use a novel lemma (Lemma 4.2.4) that allows us to relate a regret upper bound arising from application of UCB1 to a sum of lower bounds for two-armed bandit problems (and similarly in the best expert setting).

Next we explore the fully adversarial case where we make no assumptions on how the payoffs for each action are generated (in particular, they could depend on the time horizon T). This model has been extensively studied in both the best expert setting and the multi-armed bandit setting (see (Littlestone and Warmuth, 1994), (Auer et al., 2002a) and references therein). For the variant in which only a subset of the actions are available at any given time, we show that the regret of any learning algorithm must be at least $\Omega(\sqrt{Tn\log(n)})$ for the best expert setting and $\Omega(\sqrt{Tn^2})$ for the multi-armed bandit setting. We also present simple variants of algorithms in (Littlestone and Warmuth, 1994) and (Auer et al., 2002a) whose

regret is within a constant factor of the lower bound for the best expert setting, and within $\mathcal{O}(\sqrt{\log(n)})$ of the lower bound for the multi-armed bandit setting.

The fully adversarial case, however, proves to be harder, and neither algorithm is computationally efficient. To appreciate the hardness of the fully adversarial case, we prove that, unless RP = NP, any low regret algorithm that learns internally a consistent ordering over experts can not be computationally efficient (see Theorem 4.3.3). Note that this does not mean that there can be no computationally efficient, low regret algorithms for the fully adversarial case. There might exist learning algorithms that are able to achieve low regret without actually learning a consistent ordering over experts. Finding such algorithms, if they do indeed exist, remains an open problem.

4.1.1 Terminology and Conventions

We assume that there is a fixed pool of actions, $\{1, 2, ...n\}$, with n known. We will sometimes refer to an action by *expert* in the best expert setting and by *arm* or *bandit* in the multi-armed bandit setting. At each time step $t \in \{1, 2, ..., T\}$, an adversary chooses a subset $A(t) \subseteq \{1, 2, ..., n\}$ of the actions to be available. The algorithm can only choose among available actions, and only available actions receive rewards. The reward received by an available action i at time t is $r_i(t) \in [0, 1]$.

As also mentioned in Chapter 1, we will consider two models for assigning rewards to actions: a stochastic model and an adversarial model. (In contrast, the choice of the set of awake experts is always adversarial.) In the stochastic model the reward for arm i at time t, $r_i(t)$, is drawn independently from a fixed unknown distribution $P_i(\cdot)$ with bounded support and mean μ_i. In the adversarial model

we make no stochastic assumptions on how the rewards are assigned to actions. Instead, we assume that the rewards are selected by an adaptive adversary. The adversary is potentially but not necessarily randomized.

Let σ be an ordering (permutation) of the n actions, and A a subset of the actions. We denote by $\texttt{first}(A, \sigma)$ the action in A that is highest ranked in σ. That is

$$\texttt{first}(A, \sigma) = \min_{i \in \{1,2,\ldots,n\}} \sigma(1:i) \cap A \neq \emptyset,$$

where $\sigma(1:i)$ denotes the first i actions according to σ ordering.

A σ-policy corresponding to the ordering σ is the policy that selects, at each time step t, the action $\texttt{first}(A(t), \sigma)$ (i.e. available action that is highest ranked by σ). The reward of a policy σ is the reward obtained by the selected action at each time step:

$$r_\sigma(1:T) = \sum_{t=1}^{T} r_{\texttt{first}(A(t),\sigma)}(t) \tag{4.1.1}$$

Let $r_{\max\text{-}\sigma}(1:T) = \max_\sigma r_\sigma(1:T)$ ($\max_\sigma \mathbb{E}[r_\sigma(1:T)]$ in the stochastic rewards model) be the reward obtained by the best σ-policy (ordering), which is also called the benchmark. Note that in the stochastic reward model, the expectation is taken before taking the maximum over all orderings, which corresponds to the "maximum expected" reward, as opposed to the "expected maximum" reward in the adversarial setting (as is also done in the literature). We define the regret of an algorithm with respect to the best σ-policy as the expected difference between the reward obtained by the best σ-policy and the total reward of the algorithm's

chosen actions $x(1), x(2), ..., x(t)$:

$$\texttt{regret}_x(1:T) = \mathbb{E}\left[r_{\max\text{-}\sigma}(1:T) - \sum_{t=1}^{T} r_{x(t)}(t)\right], \tag{4.1.2}$$

where the expectation is taken over the algorithm's random choices and the randomness used in the reward assignment.

4.1.2 Related Work

Sequential prediction problems. The best-expert and multi-armed bandit problems correspond to special cases of our model in which every action is always available. These problems have been widely studied, and we draw on this literature to design algorithms and prove lower bounds for the generalizations considered here. The adversarial expert paradigm was introduced by Littlestone and Warmuth (1994), and Vovk (1990). Cesa-Bianchi et al. (1997) further developed this paradigm in work which gave optimal regret bounds of $\sqrt{T(\ln n)}$ and Vovk (1998) characterized the achievable regret bounds in these settings.

The multi-armed bandit model was introduced by Robbins (1952). Lai and Robbins (1985b) gave asymptotically optimal strategies for the stochastic version of bandit problem, where rewards for each arm are drawn from a fixed distribution in each time step.

Auer et al. (2002a) introduced the algorithm UCB1 (presented and analyzed in Section 2.3) and showed that the optimal regret bounds of $\mathcal{O}(\log T \cdot \sum_{i=1}^{n-1} \frac{1}{\mu_i - \mu_{i+1}})$ can be achieved uniformly over time for the stochastic bandit problem (the arms are arranged such that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$). For the adversarial version of the multi-armed bandit problem, Auer et al. (2002a) proposed the algorithm Exp3 (presented and analyzed in Section 2.4) which achieves the regret bound of $\mathcal{O}(\sqrt{Tn \log n})$,

leaving a $\sqrt{\log n}$ factor gap from the lower bound of $\Omega(\sqrt{nT})$. Recently, Audibert and Bubeck (2009) proposed a $\mathcal{O}(\sqrt{Tn})$ regret algorithm for the adversarial multi-armed bandit problem closing the sub-logarithmic gap. It is worth noting that the lower bound holds even for an oblivious adversary, one which chooses a sequence of payoff functions independently of the algorithm's choices.

Prediction with sleeping experts. Freund et al. (1997) and Blum and Mansour (2005) have analysed the sleeping experts problem in a different framework from the one we adopt here. In the model of Freund et al. (1997), as in our model, a set of awake experts is specified in each time period. The goal of the algorithm is to choose one expert in each time period so as to minimize regret against the best "mixture" of experts (which constitutes their benchmark). A mixture $\mathbf{u}$ is a probability distribution $(u_1, u_2, \ldots, u_n)$ over n experts which in time period t selects an expert according to the restriction of $\mathbf{u}$ to the set of awake experts.

In contrast, our work uses a different evaluation criterion, namely the best ordering of experts. In the special case when all experts are always awake, both evaluation criteria pick the best expert. Our "best ordering" criterion can be regarded as a degenerate case (limiting case) of the "best mixture" criterion of Freund et al. (1997) as follows. For the ordering σ, we assign probabilities $\frac{1}{Z}(1, \epsilon, \epsilon^2, \ldots, \epsilon^{n-1})$ to the sequence of experts $(\sigma(1), \sigma(2), \ldots, \sigma(n))$ where $Z = \frac{1-\epsilon^n}{1-\epsilon}$ is the normalization factor and $\epsilon > 0$ is an arbitrarily small positive constant. The only problem is that the bounds obtained from Freund et al. (1997) in this degenerate case are very weak. As $\epsilon \to 0$, their bound reduces to comparing the algorithm's performance to the ordering σ's performance only for time periods when expert $\sigma(1)$ is awake, and ignoring the time periods when $\sigma(1)$ is not awake. Therefore, a natural reduction of our problem to the problem considered by Freund et al. (1997) defeats the

purpose of giving equal importance to all time periods.

Blum and Mansour (2005) consider a generalization of the sleeping expert problem, where one has a set of *time selection functions* and the algorithm aims to have low regret with respect to every expert, according to every time selection function. It is possible to solve our regret-minimization problem (with respect to the best ordering of experts) by reducing to the regret-minimization problem solved by Blum and Mansour, but this leads to an algorithm which is neither computationally efficient nor information-theoretically optimal. We now sketch the details of this reduction. One can define a time selection function for each (ordering, expert) pair (σ, i), according to $I_{\sigma,i}(t) = 1$ if $i \preceq_\sigma j$ for all $j \in A(t)$ (that is, σ chooses i in time period t if $I_{\sigma,i}(t) = 1$). The regret can now be bounded, using Blum and Mansour's analysis, as

$$\sum_{i=1}^{n} \mathcal{O}\left(\sqrt{T_i \log(n \cdot n! \cdot n)} + \log(n! \cdot n^2)\right) = \mathcal{O}\left(\sqrt{Tn^2 \log n} + n^2 \log n\right).$$

This algorithm takes exponential time (due to the exponential number of time selection functions) and gives a regret bound of $\mathcal{O}(\sqrt{Tn^2 \log n})$ against the best ordering, a bound which we improve in Section 4.3 using a different algorithm which also takes exponential time but is information-theoretically optimal. (Of course, Blum and Mansour were designing their algorithm for a different objective, not trying to get low regret with respect to best ordering. Our improved bound for regret with respect to the best ordering does not imply an improved bound for experts learning with time selection functions.)

Langford and Zhang (2007) presents an algorithm called the *Epoch-Greedy algorithm* for bandit problems with side information. This is a generalization of the multi-armed bandit problem in which the algorithm is supplied with a piece of *side information* in each time period before deciding which action to play. Given

a hypothesis class $\mathcal{H}$ of functions mapping side information to actions, the Epoch-Greedy algorithm achieves low regret against a sequence of actions generated by applying a single function $h \in \mathcal{H}$ to map the side information in every time period to an action. (The function h is chosen so that the resulting sequence has the largest possible total payoff.) The stochastic case of our problem is reducible to theirs, by treating the set of available actions, $A(t)$, as a piece of side information and considering the hypothesis class $\mathcal{H}$ consisting of functions h_σ, for each total ordering σ of the set of actions, such that $h_\sigma(A)$ selects the element of A which appears first in the ordering σ. The regret bound in Langford and Zhang (2007) is expressed implicitly in terms of the expected regret of an empirical reward maximization estimator, which makes it difficult to compare this bound with ours. Instead of pursuing this reduction from our problem to the contextual bandit problem in Langford and Zhang (2007), we propose a very simple bandit algorithm for the stochastic setting with an explicit regret bound that is provably information-theoretically optimal.

4.2 Stochastic Model of Rewards

We first explore the stochastic rewards model, where the reward for action i at each time step is drawn independently from a fixed unknown distribution $P_i(\cdot)$ with mean μ_i. For simplicity of presentation, throughout this section we assume that $\mu_1 > \mu_2 > \cdots > \mu_n$. That is, the lower numbered actions are better than the higher numbered actions. Let $\Delta_{i,j} = \mu_i - \mu_j$ for all $i < j$ be the increase in the expected reward of expert i over expert j.

We present optimal (up to a constant factor) algorithms for both the best expert and the multi-armed bandit setting. Both algorithms are natural extensions of

algorithms for the all-awake problem to the sleeping-experts problem. The analysis of the algorithms, however, is not a straightforward extension of the analysis for the all-awake problem and new proof techniques are required.

4.2.1 Best Expert Setting

In this section we study the best expert setting with stochastic rewards. We provide an algorithm and prove matching (up to a constant factor) information-theoretic lower bounds on the regret of any algorithm.

Upper Bound (Algorithm: FTAL)

To get an upper bound on regret we adapt the "follow the leader" algorithm (Hannan, 1957; Kalai and Vempala, 2005) to the sleeping experts setting: at each time step the algorithm chooses the awake expert that has the highest average payoff, where the average is taken over the time steps when the expert was awake. If an expert is awake for the first time, then the algorithm chooses it. (If there is more than one such expert, then the algorithm chooses one of them arbitrarily.) The pseudocode for the algorithm is shown in Algorithm 4.1. The algorithm is called **F**ollow **T**he **A**wake **L**eader (FTAL for short).

The performance guarantee of the algorithm FTAL is presented in the following theorem.

Theorem 4.2.1. *Let* $\Delta_{i,i+1} > 0$ *for* $i = 1, 2, \ldots, n-1$. *Then* FTAL *algorithm has a regret of at most*

$$\sum_{i=1}^{n-1} \frac{32}{\Delta_{i,i+1}},$$

with respect to the best ordering.

Initialize $z_i = 0$ and $n_i = 0$ for all $i \in [n]$.

FOR $t = 1$ to T

  IF $\exists j \in A(t)$ s.t. $n_j = 0$

    Play expert $x(t) = j$

  ELSE

    Play expert $x(t) = \arg\max_{i \in A(t)} \left(\frac{z_i}{n_i}\right)$.

  Observe payoff $r_i(t)$ for all $i \in A(t)$.

  $z_i \leftarrow z_i + r_i(t)$ for all $i \in A(t)$

  $n_i \leftarrow n_i + 1$ for all $i \in A(t)$

Figure 4.1: Follow-the-awake-leader (FTAL) algorithm for the sleeping experts problem with a stochastic adversary.

Note that we are only considering problem instances in which different arms have different average payoffs. Also note that as $\Delta_{i,i+1}$ gets close to 0, the regret bound become vacuous. A general result will be proved in Theorem 4.2.6 which will take care of both these restrictions, and the above theorem follows as a corollary to Theorem 4.2.6 by setting $\epsilon = 0$.

The above theorem follows immediately from the following pair of lemmas. The second of these lemmas will also be used in Section 4.2.2.

Lemma 4.2.2. *Let $\Delta_{i,i+1} > 0$ for $i = 1, 2, \ldots, n-1$. Then the* FTAL *algorithm has a regret of at most*

$$\sum_{j=2}^{n} \sum_{i=1}^{j-1} \frac{8}{\Delta_{i,j}^2} (\Delta_{i,i+1} + \Delta_{j-1,j})$$

with respect to the best ordering.

Proof. Let $n_i(t)$ be the number of times expert i has been awake until time t. Let $\hat{\mu}_i(t)$ be expert i's average payoff until time t. The Azuma-Hoeffding Inequality

(Azuma, 1967; Hoeffding, 1963) says that

$$\mathbb{P}[n_j(t)\hat{\mu}_j(t) > n_j(t)\mu_j + n_j(t)\Delta_{i,j}/2] \leq e^{-\frac{n_j(t)^2\Delta_{i,j}^2}{8\cdot n_j(t)}} = e^{-\frac{\Delta_{i,j}^2 n_j(t)}{8}},$$

and

$$\mathbb{P}[n_i(t)\hat{\mu}_i(t) < n_i(t)\mu_i - n_i(t)\Delta_{i,j}/2] \leq e^{-\frac{n_i(t)^2\Delta_{i,j}^2}{8\cdot n_i(t)}} = e^{-\frac{\Delta_{i,j}^2 n_i(t)}{8}}.$$

Let us say that the FTAL algorithm suffers an (i,j)-*anomaly of type 1* at time t if $x_t = j$ and $\hat{\mu}_j(t) - \mu_j > \Delta_{i,j}/2$; note that the definition does not require expert i to be awake at time t. Define $i^*(t)$ to be the optimal expert at time t (lowest indexed expert in $A(t)$). Let us say that FTAL suffers an (i,j)-*anomaly of type 2* at time t if $i^*(t) = i$ and $\mu_i - \hat{\mu}_i(t) > \Delta_{i,j}/2$; note again that the definition does not require expert j to be awake at time t. Note that when FTAL picks a strategy $x(t) = j \neq i = i^*(t)$, it suffers an (i,j)-anomaly of type 1 or 2, or possibly both. We will denote the event of an (i,j)-anomaly of type 1 (resp. type 2) at time t by $\mathcal{E}_{i,j}^{(1)}(t)$ (resp. $\mathcal{E}_{i,j}^{(2)}(t)$), and we will use $M_{i,j}^{(1)}$, resp. $M_{i,j}^{(2)}$, to denote the total number of (i,j)-anomalies of types 1 and 2, respectively. We can bound the expected value of $M_{i,j}^{(1)}$ by

$$\begin{aligned}
\mathbb{E}[M_{i,j}^{(1)}] &\leq \sum_{t=1}^{\infty} e^{-\frac{\Delta_{i,j}^2 n_j(t)}{8}} \mathbf{1}\{j \in A(t)\} && (4.2.1)\\
&\leq \sum_{n=1}^{\infty} e^{-\frac{\Delta_{i,j}^2 n}{8}} && (4.2.2)\\
&= \frac{1}{e^{\Delta_{i,j}^2/8} - 1} \leq \frac{8}{\Delta_{i,j}^2},
\end{aligned}$$

where line (4.2.2) is justified by observing that distinct nonzero terms in (4.2.1) have distinct values of $n_j(t)$. The expectation of $M_{i,j}^{(2)}$ is also bounded by $8/\Delta_{i,j}^2$, via an analogous argument.

Recall that $A(t)$ denotes the set of awake experts at time t, $x(t) \in A(t)$ denotes the algorithm's choice at time t, and $r_i(t)$ denotes the payoff of expert i at time t

(which is distributed according to $P_i(\cdot)$). Recall that $i^*(t) \in A(t)$ is the optimal expert at time t (i.e., the lowest-numbered element of $A(t)$). We are now ready to bound the regret of the FTAL algorithm. A very crucial observation that we make next is that when arm $i^*(t)$ is the optimal arm in round t and arm $x(t) \neq i^*(t)$ is picked by the algorithm, one of the following two events must have happened: either the observed reward of arm $i^*(t)$ is much *smaller* than its actual mean $\mu_{i^*(t)}$, or the observed reward of arm $x(t)$ is much *larger* than its actual mean $\mu_{x(t)}$. The first one corresponds to an $(i^*(t), x(t))$-anomaly of type 2, and the second one corresponds to an $(i^*(t), x(t))$-anomaly of type 1. We split the regret according to this classification, and bound each term in turn.

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\left(r_{i^*(t)}(t) - r_{x(t)}(t)\right)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\Delta_{i^*(t),x(t)}\right] = \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}^{(1)}_{i^*(t),x(t)}(t) \vee \mathcal{E}^{(2)}_{i^*(t),x(t)}(t)\right\}\Delta_{i^*(t),x(t)}\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}^{(1)}_{i^*(t),x(t)}(t)\right\}\Delta_{i^*(t),x(t)}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}^{(2)}_{i^*(t),x(t)}(t)\right\}\Delta_{i^*(t),x(t)}\right]. \quad (4.2.3)
\end{aligned}$$

With the convention that $\Delta_{i,j} = 0$ for $j \leq i$, the first term in (4.2.3) can be bounded as follows.

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}^{(1)}_{i^*(t),x(t)}(t)\right\}\Delta_{i^*(t),x(t)}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n}\mathbf{1}\left\{\mathcal{E}^{(1)}_{i^*(t),j}(t)\right\}\Delta_{i^*(t),j}\right] \quad \text{(Since the event } \mathcal{E}^{(1)}_{i^*(t),j}(t) \\
&\qquad\qquad \text{occurs only for } j = x(t).) \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n}\mathbf{1}\left\{\mathcal{E}^{(1)}_{i^*(t),j}(t)\right\}\sum_{i=i^*(t)}^{j-1}\Delta_{i,i+1}\right] \quad (4.2.4) \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n}\sum_{i=i^*(t)}^{j-1}\mathbf{1}\left\{\mathcal{E}^{(1)}_{i,j}(t)\right\}\Delta_{i,i+1}\right] \quad \text{(Since} \\
&\qquad\qquad \mathbf{1}\left\{\mathcal{E}^{(1)}_{i_1,j}(t)\right\} \leq \mathbf{1}\left\{\mathcal{E}^{(1)}_{i_2,j}(t)\right\} \\
&\qquad\qquad \text{for } i_1 \leq i_2 < j.)
\end{aligned}$$

$$
\begin{aligned}
&\leq \mathbb{E}\left[\sum_{j=2}^{n}\sum_{i=1}^{j-1}\Delta_{i,i+1}\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}_{i,j}^{(1)}(t)\right\}\right] \\
&= \sum_{j=2}^{n}\sum_{i=1}^{j-1}\Delta_{i,i+1}\mathbb{E}[M_{i,j}^{(1)}] \\
&\leq \sum_{1\leq i<j\leq n}\frac{8}{\Delta_{i,j}^2}\Delta_{i,i+1}.
\end{aligned}
$$

Similarly, the second term in (4.2.3) can be bounded by

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}_{i^*(t),x(t)}^{(2)}(t)\right\}\Delta_{i^*(t),x(t)}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n-1}\mathbf{1}\left\{\mathcal{E}_{i,x(t)}^{(2)}(t)\right\}\Delta_{i,x(t)}\right] \quad \text{(Since event } \mathcal{E}_{i,x(t)}^{(2)}(t) \text{ occurs only for } i=i^*(t).\text{)} \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n-1}\mathbf{1}\left\{\mathcal{E}_{i,x(t)}^{(2)}(t)\right\}\sum_{j=i+1}^{x(t)}\Delta_{j-1,j}\right] \qquad (4.2.5) \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n-1}\sum_{j=i+1}^{x(t)}\mathbf{1}\left\{\mathcal{E}_{i,j}^{(2)}(t)\right\}\Delta_{j-1,j}\right] \quad \text{(For } i<j_1\leq j_2, \ \mathbf{1}\left\{\mathcal{E}_{i,j_1}^{(2)}(t)\right\}\geq\mathbf{1}\left\{\mathcal{E}_{i,j_2}^{(2)}(t)\right\}.\text{)} \\
&\leq \mathbb{E}\left[\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\Delta_{j-1,j}\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}_{i,j}^{(2)}(t)\right\}\right] \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\Delta_{j-1,j}\mathbb{E}[M_{i,j}^{(2)}] \\
&\leq \sum_{1\leq i<j\leq n}\frac{8}{\Delta_{i,j}^2}\Delta_{j-1,j}
\end{aligned}
$$

Adding the two bounds gives the statement of the lemma. □

Before presenting the next lemma that will finish the proof of Theorem 4.2.1, let us make the following definition which will be useful in the proof.

Definition 4.2.3. For an expert j and $y \geq 0$, let $i_y(j)$ be the minimum numbered expert $i \leq j$ such that $\Delta_{i,j}$ is no more than y. That is

$$
i_y(j) := \arg\min\{i : i \leq j, \Delta_{i,j} \leq y\}.
$$

For an expert i, and $y \geq 0$, let $j_y(i)$ be the maximum numbered expert $j \geq i$ such that $\Delta_{i,j}$ is no more than y. That is

$$j_y(i) := \arg\max\{j : j \geq i, \Delta_{i,j} \leq y\}.$$

Now we are ready to present our next lemma.

Lemma 4.2.4. *Let $\Delta_{i,i+1} > 0$ for $i = 1, 2, \ldots, n-1$. Then*

$$\sum_{1 \leq i < j \leq n} \Delta_{i,j}^{-2} \Delta_{i,i+1} \leq 2 \sum_{j=2}^{n} \Delta_{j-1,j}^{-1} \quad \textit{and} \quad \sum_{1 \leq i < j \leq n} \Delta_{i,j}^{-2} \Delta_{j-1,j} \leq 2 \sum_{i=1}^{n-1} \Delta_{i,i+1}^{-1}.$$

Note that this lemma is very important from a technical point of view in the proof of the regret bound for FTAL, but does not have a direct bearing on the intuitive understanding of the algorithm.

Note that Lemma 4.2.4 combined with Lemma 4.2.2 finishes the proof of Theorem 4.2.1. Instead of proving the lemma above, we will prove a slight generalization (that will be useful in taking care of "small $\Delta_{i,i+1}$'s"), and the lemma above will follow as a special case by putting $\epsilon = 0$.

Let us first motivate the generalization. The left hand side of the first inequality in Lemma 4.2.4 can also be written as $\sum_{1 \leq i < j \leq n : \Delta_{i,j} > 0} \Delta_{i,j}^{-2} \Delta_{i,i+1}$, since the condition $\Delta_{i,j} > 0$ is vacuous (we are assuming in the statement of the lemma that $\Delta_{i,j} > 0$ for $i < j$). Instead of putting an upper bound on $\sum_{1 \leq i < j \leq n : \Delta_{i,j} > 0} \Delta_{i,j}^{-2} \Delta_{i,i+1}$, we will relax the condition $\Delta_{i,j} > 0$ to $\Delta_{i,j} > \epsilon$ for some $\epsilon \geq 0$ and prove an upper bound on $\sum_{1 \leq i < j \leq n : \Delta_{i,j} > \epsilon} \Delta_{i,j}^{-2} \Delta_{i,i+1}$. Let us present the general case.

Lemma 4.2.5. *For $\epsilon \geq 0$,*

$$\sum_{1 \leq i < j \leq n : \Delta_{i,j} > \epsilon} \Delta_{i,j}^{-2} \Delta_{i,i+1} \leq 2 \sum_{j=j_0(1)+1}^{n} \max\{\epsilon, \Delta_{i_0(j)-1,i_0(j)}\}^{-1} \quad \textit{and}$$

$$\sum_{1\le i<j\le n:\Delta_{i,j}>\epsilon} \Delta_{i,j}^{-2}\Delta_{j-1,j} \le 2\sum_{i=1}^{j_0(n)-1} \max\{\epsilon, \Delta_{j_0(i),j_0(i)+1}\}^{-1}.$$

Recall from Definition 4.2.3 that if $\Delta_{i,j} > 0$ for $i < j$, then $j_0(i) = i$ for all i and $i_0(j) = j$ for all j, and the above lemma reduces to Lemma 4.2.4 by taking $\epsilon = 0$.

Proof. It suffices to prove the first of the two inequalities stated in the lemma; the second follows from the first by replacing each μ_i with $1 - \mu_i$, which has the effect of replacing $\Delta_{i,j}$ with $\Delta_{n+1-j,n+1-i}$.

For a fixed $i \in [n]$, we write $\sum_{j:j>i,\Delta_{i,j}>\epsilon} \Delta_{i,j}^{-2}$ as follows.

$$\begin{aligned}
\sum_{j:j>i,\Delta_{i,j}>\epsilon} \Delta_{i,j}^{-2} &= \sum_{j=2}^{n} \mathbf{1}\left\{j > i, \Delta_{i,j} > \epsilon\right\} \Delta_{i,j}^{-2} \qquad (4.2.6)\\
&= \int_{x=0}^{\infty} \left|\left\{j : j > i, \Delta_{i,j} > \epsilon, \Delta_{i,j}^{-2} \ge x\right\}\right| dx\\
&= \int_{x=0}^{\infty} \left|\left\{j : \epsilon < \Delta_{i,j} \le x^{-1/2}\right\}\right| dx \quad (\Delta_{i,j} > \epsilon \text{ implies } j > i.)\\
&= -2\int_{y=\infty}^{0} |\{j : \epsilon < \Delta_{i,j} \le y\}|\, y^{-3}\, dy \quad \text{(Changing the variable of integration } x^{-1/2} = y.\text{)}\\
&= 2\int_{y=0}^{\infty} |\{j : \epsilon < \Delta_{i,j} \le y\}|\, y^{-3}\, dy. \qquad (4.2.7)
\end{aligned}$$

Now we can write the following chain of inequalities. (Note that the best (highest payoff) expert is indexed as 1, and lowest payoff is indexed n.)

$$\begin{aligned}
&\sum_{i=1}^{n-1} \sum_{j\in\{i+1,i+2,\ldots,n\},\Delta_{i,j}>\epsilon} \Delta_{i,j}^{-2}\Delta_{i,i+1}\\
&= \sum_{i=1}^{n-1} \Delta_{i,i+1} \sum_{j:j>i,\Delta_{i,j}>\epsilon} \Delta_{i,j}^{-2} \qquad (4.2.8)
\end{aligned}$$

$$= 2\sum_{i=1}^{n-1} \Delta_{i,i+1} \left(\int_{y=0}^{\infty} |\{j : \epsilon < \Delta_{i,j} \leq y\}| \, y^{-3} \, dy \right) \quad \text{(From (4.2.7).)}$$

$$= 2\int_{y=0}^{\infty} y^{-3} \left(\sum_{i=1}^{n-1} \Delta_{i,i+1} \cdot |\{j : \epsilon < \Delta_{i,j} \leq y\}| \right) dy \quad \text{(Changing the order of integration and summation.)}$$

$$= 2\int_{y=0}^{\infty} y^{-3} \left(\sum_{i=1}^{n-1} \Delta_{i,i+1} \sum_{j=i+1}^{n} \mathbf{1}\{\epsilon < \Delta_{i,j} \leq y\} \right) dy \quad \text{(Expanding } |\{\cdot\}| \text{ into sum of } \mathbf{1}\{\cdot\}.)$$

$$= 2\int_{y=0}^{\infty} y^{-3} \left(\sum_{j=2}^{n} \sum_{i=1}^{j-1} \Delta_{i,i+1} \mathbf{1}\{\epsilon < \Delta_{i,j} \leq y\} \right) dy \quad \text{(Changing the order of summation.)}$$

$$= 2\sum_{j=2}^{n} \int_{y=0}^{\infty} y^{-3} \left(\sum_{i=1}^{j-1} \Delta_{i,i+1} \mathbf{1}\{\epsilon < \Delta_{i,j} \leq y\} \right) dy \quad \text{(Changing the order of summation and integration.)}$$

$$= 2\sum_{j=2}^{n} \int_{y=\epsilon}^{\infty} y^{-3} \left(\sum_{i=1}^{j-1} \Delta_{i,i+1} \mathbf{1}\{\epsilon < \Delta_{i,j} \leq y\} \right) dy \quad \text{(For } y < \epsilon\text{, the integrand is 0.)}$$

$$= 2\sum_{j=2}^{n} \int_{y=\epsilon}^{\infty} y^{-3} \left(\sum_{i=i_y(j)}^{i_\epsilon(j)-1} \Delta_{i,i+1} \right) dy \quad \text{(Use Definition 4.2.3.)}$$

$$= 2\sum_{j=2}^{n} \int_{y=\epsilon}^{\infty} y^{-3} \left(\mu_{i_y(j)} - \mu_{i_{\epsilon(j)}} \right) dy$$

Now, we need a little care in manipulating this expression. Let us consider two cases: (i) $\mu_{i_\epsilon(j)} = \mu_{i_0(j)}$, which means that there is no arm with mean in $(\mu_j, \mu_j + \epsilon]$, and (ii) $\mu_{i_{\epsilon(j)}} > \mu_{i_0(j)}$, which means that there is some arm with mean in $(\mu_j, \mu_j + \epsilon]$. In the first case, $\mu_{i_y(j)} - \mu_{i_\epsilon(j)}$ is zero whenever $y < \Delta_{i_0(j)-1,i_0(j)}$, so the lower limit of the integration can be changed to $\Delta_{i_0(j)-1,i_0(j)}$. In the second case, no special care needs to be taken. Note that in both cases, $\mu_{i_y(j)} - \mu_{i_\epsilon(j)} \leq y$. Also note that for j such that $\mu_j = \mu_1$, the difference $\mu_{i_y(j)} - \mu_{i_\epsilon(j)}$ is always zero (both terms being equal to μ_1. So, we can change the lower limit of the outer sum to start from $j_0(1) + 1$ (the first arm which has mean lower than the mean of the first arm).)

$$
\begin{aligned}
&\leq 2 \sum_{j=j_0(1)+1}^{n} \left(\mathbf{1}\left\{\Delta_{i_0(j)-1,i_0(j)} > \epsilon\right\} \int_{y=\Delta_{i_0(j)-1,i_0(j)}}^{\infty} y^{-2}\, dy + \mathbf{1}\left\{\Delta_{i_0(j)-1,i_0(j)} \leq \epsilon\right\} \int_{y=\epsilon}^{\infty} y^{-2}\, dy \right) \\
&= 2 \sum_{j=j_0(1)+1}^{n} \left(\mathbf{1}\left\{\Delta_{i_0(j)-1,i_0(j)} > \epsilon\right\} \left(\Delta_{i_0(j)-1,i_0(j)}\right)^{-1} + \mathbf{1}\left\{\Delta_{i_0(j)-1,i_0(j)} \leq \epsilon\right\} (\epsilon)^{-1} \right) \\
&= 2 \sum_{j=j_0(1)+1}^{n} \left(\max\left\{\epsilon, \Delta_{i_0(j)-1,i_0(j)}\right\}\right)^{-1}
\end{aligned}
$$

This concludes the proof of the lemma. □

Remarks for small $\Delta_{i,i+1}$ Note that the upper bound stated in Theorem 4.1 become very large when $\Delta_{i,i+1}$ is very small for some i. Indeed, when mean payoffs of all experts are equal, $\Delta_{i,i+1} = 0$ for all i and upper bound becomes trivial, while the algorithm does well (picking any expert is as good as any other). We suggest a slight modification of the proof to take care of such case.

Let $\epsilon > 0$ be fixed (the original theorem corresponds to the case $\epsilon = 0$). Recall the definition of $i_\epsilon(j)$ and $j_\epsilon(i)$ from Definition 4.2.3. Note that the three conditions: (1) $i < i_\epsilon(j)$, (2) $j > j_\epsilon(i)$, and (3) $\Delta_{i,j} > \epsilon$ are equivalent. The idea in this new analysis is to "identify" experts that have means within ϵ of each other. (We cannot just make equivalence classes based on this, since the relation of "being

within ϵ of each other" is not an equivalence relation.)

Lemma 4.2.2 can be modified to prove that the regret of the algorithm is bounded by

$$2\epsilon T + \sum_{\substack{1 \le i < j \le n, \\ \Delta_{i,j} > \epsilon}} \frac{8}{\Delta_{i,j}^2}(\Delta_{i,i+1} + \Delta_{j-1,j}).$$

This can be seen by rewriting Equation (4.2.4) as

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n} \mathbf{1}\left\{\mathcal{E}^{(1)}_{i^*(t),j}(t)\right\} \sum_{i=i^*(t)}^{i_\epsilon(j)-1} \Delta_{i,i+1}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n} \mathbf{1}\left\{\mathcal{E}^{(1)}_{i^*(t),j}(t)\right\} \sum_{i=i_\epsilon(j)}^{j-1} \Delta_{i,i+1}\right]$$

and noting that the second term is at most

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n} \mathbf{1}\left\{\mathcal{E}^{(1)}_{i^*(t),j}(t)\right\} \epsilon\right] = \mathbb{E}\left[\epsilon \sum_{t=1}^{T} 1\right] = \epsilon T,$$

since only one of the events $\mathcal{E}^{(1)}_{i^*(t),j}(t)$ (corresponding to $j = x(t)$) can occur for each t. Equation (4.2.5) can be similarly modified by splitting the summation $j = i + 1 \cdots x(t)$ to $j = i + 1 \cdots j_\epsilon(i)$ and $j = j_\epsilon(i) + 1 \cdots x(t)$.

To upper bound the regret by the sum of inverses of $\Delta_{i,i+1}$, we can use Lemma 4.2.5. With these modifications to the proof, we have established the following variant of Theorem 4.2.1. Note that the result of Theorem 4.2.1 can be seen to be a special case of the theorem below by setting $\epsilon = 0$.

Theorem 4.2.6. *For every $\epsilon \ge 0$, the* FTAL *algorithm has a regret of at most*

$$2\epsilon T + \sum_{j=j_0(1)+1}^{n} \frac{16}{\max\{\epsilon, \Delta_{i_0(j)-1,i_0(j)}\}} + \sum_{i=1}^{j_0(n)-1} \frac{16}{\max\{\epsilon, \Delta_{j_0(i),j_0(i)+1}\}}.$$

with respect to the best ordering.

Lower Bound

In this section, assuming that the means μ_i are bounded away from 0 and 1, we prove that FTAL's regret presented in the section above is optimal (up to constant

factors). This is done by showing the following lower bound on the regret guarantee of any algorithm. Let Bernoulli(p) denote the Bernoulli distribution with mean p. We use $\mathsf{KL}(p;q)$ to denote the KL-divergence of two distributions, and for the case of Bernoulli distributions with means μ and μ', we use the notation $\mathsf{KL}(\mu;\mu')$ instead of writing a somewhat more wordy notation $\mathsf{KL}(\text{Bernoulli}(\mu), \text{Bernoulli}(\mu'))$. Please refer to Karp and Kleinberg (2007b) and Cover and Thomas (1999) for an introduction to KL-divergence.

Lemma 4.2.7. *Let $P_i = \text{Bernoulli}(\mu_i)$ for $i = 1, 2, \ldots, n$ be the payoff distributions with $\mu_i \in (\alpha, \beta)$ for some $0 < \alpha < \beta < 1$ (μ_i's can be relaxed to lie in the* closed *interval $[\alpha, \beta]$). Let ϕ be any algorithm for the stochastic best expert model. Then, there is an input instance with n arms endowed with some permutation of the aforementioned distributions $(P_1, P_2, \ldots, P_n)$, such that the regret of ϕ up to time T is at least*

$$\Omega\left(\sum_{i=1}^{n-1} \frac{1}{\Delta_{i,i+1}}\right),$$

whenever $T \geq T_0$, where T_0 is a function of n, $(\mu_1, \mu_2, \ldots, \mu_n)$, α, and β.

To prove this lemma, we first prove its special case for the case of two experts.

Lemma 4.2.8. *Let $P_i = \text{Bernoulli}(\mu_i)$ for $i = 1, 2$ be payoff distribution with $\mu_1, \mu_2 \in (\alpha, \beta)$, $\mu_1 > \mu_2$, and $0 < \alpha < \beta < 1$. Let ϕ be an online algorithm for the stochastic best expert problem with two experts. Consider two instances I_1 and I_2 for the stochastic best expert setting: In both instances, there are two experts namely L and R; in I_1, (L, R) are endowed with reward distributions (P_1, P_2) and in I_2, they are endowed with (P_2, P_1). Then the regret of algorithm ϕ on at least one of I_1 or I_2 is*

$$\Omega\left(\delta^{-1}\right),$$

whenever $T \geq T_0$, where $\delta = \mu_1 - \mu_2$, T_0 is a function of (μ_1, μ_2), α, and β, and the constants inside the $\Omega(\cdot)$ may depend on α, β.

Proof. Let us define some joint distributions: p is the *joint* distribution in which both experts have payoff distribution P_1, q_L is the distribution in which they have payoff distributions (P_1, P_2) (left is better), and q_R is the distribution in which they have payoff distributions (P_2, P_1) (right expert is better).

Let $T_0 = \frac{c}{\delta^2}$ for $c = \frac{\min\{\alpha(1-\alpha), \beta(1-\beta)\}}{25}$, and $T \geq T_0$. We will prove that if ϕ runs for T rounds, then for one the instances q_L or q_R, it will suffer at least $\Omega(\delta^{-1})$ regret.

Let us define the following events: $E^L(t)$ is true if ϕ picks L at time t, and similarly $E^R(t)$.

We denote by $p^t(\cdot)$ the distribution induced by ϕ on the t-step histories, where the distribution of rewards in each time period is $p(\cdot)$. Similarly for $q^t(\cdot)$. We have $p^t[E^L(t)] + p^t[E^R(t)] = 1$. Therefore, for every t, there exists $M \in \{L, R\}$ such that $p^t[E^M(t)] \geq 1/2$. Similarly, there exists $M \in \{L, R\}$ such that

$$\left| \left\{ t : 1 \leq t \leq T, \quad p^t[E^M(t)] \geq \frac{1}{2} \right\} \right| \geq \frac{T}{2}.$$

Without loss of generality, assume that $M = L$. Now assume the algorithm faces the input distribution q_R, and define $q = q_R$. Using $\mathsf{KL}(\cdot;\cdot)$ to denote the KL-divergence of two distributions, we have

$$\begin{aligned} \mathsf{KL}(p^t; q^t) \leq \mathsf{KL}(p^T; q^T) = T \cdot \mathsf{KL}(p; q) = c\delta^{-2} \cdot \mathsf{KL}(\mu_1; \mu_2) \\ \leq c\delta^{-2} \cdot \frac{\delta^2}{2\min\{\alpha(1-\alpha), \beta(1-\beta)\}} \leq \frac{1}{50}, \end{aligned}$$

by the choice of c.

Karp and Kleinberg (2007b) prove the following lemma. If there is an event E with $p(E) \geq 1/3$ and $q(E) < 1/3$, then

$$\mathsf{KL}(p; q) \geq \frac{1}{3} \ln \left(\frac{1}{3q(E)} \right) - \frac{1}{e}. \tag{4.2.9}$$

We have that for at least $T/2$ values of t, $p^t(E^L(t)) \geq 1/3$ (it is actually at least $1/2$). In such time steps, we either have $q^t(E^L(t)) \geq 1/3$ or the lemma applies, yielding

$$\frac{1}{50} \geq \mathsf{KL}(p^t; q^t) \geq \frac{1}{3} \ln \left(\frac{1}{q^t(E^L(t))} \right) - \frac{1}{e}.$$

This gives $q^t(E^L(t)) \geq \frac{1}{10}$. Therefore, the regret of the algorithm in time period t is at least

$$\mu_1 - \left(\frac{9}{10}\mu_1 + \frac{1}{10}\mu_2 \right) \geq \frac{1}{10}\delta.$$

Since $T = \Omega(\delta^{-2})$, we have that the regret is at least

$$\frac{1}{10}\delta \cdot \Omega(\delta^{-2}) = \Omega(\delta^{-1}).$$

This finishes the proof of the lower bound for two experts. We next prove the lower bound for n experts. □

Proof of Lemma 4.2.7: Let us group experts in pairs of 2 as $(2i-1, 2i)$ for $i = 1, 2, \ldots, \lfloor n/2 \rfloor$. Apply the two-expert lower bound from Lemma 4.2.8 by creating a series of time steps when $A(t) = \{2i-1, 2i\}$ for each i. (We need a sufficiently large time horizon — namely $T \geq \sum_{i=1}^{\lfloor n/2 \rfloor} c\Delta_{2i-1,2i}^{-2}$ — in order to apply the lower bound to all $\lfloor n/2 \rfloor$ two-expert instances.) The total regret suffered by any algorithm is the sum of regret suffered in the independent $\lfloor n/2 \rfloor$ instances defined above. Using the lower bound from Lemma 4.2.8, we get that the regret suffered by any algorithm is at least

$$\sum_{i=1}^{\lfloor n/2 \rfloor} \Omega \left(\frac{1}{\Delta_{2i-1,2i}} \right).$$

Similarly, if we group the experts in pairs according to $(2i, 2i+1)$ for $i = 1, 2, \ldots, \lfloor n/2 \rfloor$, then we get a lower bound of

$$\sum_{i=1}^{\lfloor n/2 \rfloor} \Omega\left(\frac{1}{\Delta_{2i,2i+1}}\right).$$

Since both of these are lower bounds, so is their average, which is

$$\frac{1}{2}\sum_{i=1}^{n-1} \Omega\left(\frac{1}{\Delta_{i,i+1}}\right) = \Omega\left(\sum_{i=1}^{n-1} \Delta_{i,i+1}^{-1}\right).$$

This proves the lemma. □

4.2.2 Multi-Armed Bandit Setting

We now turn our attention to the multi-armed bandit setting against a stochastic adversary. We first present a variant of the UCB1 algorithm (Auer et al., 2002a), and then present a matching lower bound based on an idea from Lai and Robbins (1985b).

Upper Bound (Algorithm: AUER)

Here the optimal algorithm is again a natural extension of the UCB1 algorithm (Auer et al., 2002a) to the sleeping-bandits case. In a nutshell, the algorithm keeps track of the running average of payoffs received from each arm, and also a confidence interval of width (radius) $\rho_j(t) = \sqrt{\frac{8 \ln t}{n_j(t)}}$ around arm j, where t is the current time interval and $n_j(t)$ is the number of times j's payoff has been observed (number of times arm j has been played). At time t, if an arm becomes available for the first time then the algorithm chooses it. Otherwise the algorithm optimistically picks the arm with highest "upper estimated reward" (or "upper confidence bound" in UCB1 terminology) among the available arms. That is, it picks the arm $j \in A(t)$ with maximum $\hat{\mu}_j(t) + \rho_j(t)$ where $\hat{\mu}_j(t)$ is the mean of the

Initialize $z_i = 0$ and $n_i = 0$ for all $i \in [n]$.

FOR $t = 1$ to T

  IF $\exists j \in A(t)$ s.t. $n_j = 0$

    Play arm $x(t) = j$

  ELSE

    Play arm $x(t) = \arg\max_{i \in A_t} \left(\frac{z_i}{n_i} + \sqrt{\frac{8 \log t}{n_i}} \right)$

  Observe payoff $r_{x(t)}(t)$ for arm $x(t)$

  $z_{x(t)} \leftarrow z_{x(t)} + r_{x(t)}(t)$

  $n_{x(t)} \leftarrow n_{x(t)} + 1$

Figure 4.2: The AUER algorithm for the sleeping bandit problem with a stochastic adversary.

observed rewards of arm j up to time t, and $\rho_j(t) = \sqrt{\frac{8 \ln t}{n_j(t)}}$ is the width of the confidence interval around arm j at time t. The algorithm is shown in Figure 4.2. The algorithm is called **A**wake **U**pper **E**stimated **R**eward (AUER).

We first need to state a claim about the confidence intervals that we are using.

Lemma 4.2.9. *With the definition of* $n_i(t)$, μ_i, $\hat{\mu}_i(t)$, *and* $\rho_i(t) = \sqrt{\frac{8 \ln t}{n_i(t)}}$ *the following holds for all* $1 \leq i \leq n$ *and* $1 \leq t \leq T$*:*

$$\mathbb{P}\Big[\mu_i \in [\hat{\mu}_i(t) - \rho_i(t), \hat{\mu}_i(t) + \rho_i(t)]\Big] = \mathbb{P}\Big[\hat{\mu}_i(t) \in [\mu_i - \rho_i(t), \mu_i + \rho_i(t)]\Big] \geq 1 - \frac{1}{t^4}.$$

Proof. The equality follows since the two events are the same. The proof of inequality is an application of Chernoff-Hoeffding bounds, and follows from (Auer et al., 2002a, pp. 242–243). □

Theorem 4.2.10. *For problem instances with* $\Delta_{i,i+1} > 0$ *for* $i = 1, 2, \ldots, n-1$,

the regret of the AUER *algorithm is at most*

$$(66 \ln T + \mathcal{O}(1)) \cdot \sum_{i=1}^{n-1} \frac{1}{\Delta_{i,i+1}}.$$

up to time T.

The theorem follows immediately from the following lemma and Lemma 4.2.4. Note that we are only considering problem instances in which different arms have different means. This restriction will be removed at the end of this section, where we present a general bound, and the above theorem will follows as a special case of the general result.

Lemma 4.2.11. *For problem instances with* $\Delta_{i,i+1} > 0$ *for* $i = 1, 2, \ldots, n-1$*, the* AUER *algorithm has a regret of at most*

$$(33 \ln T + \mathcal{O}(1)) \cdot \sum_{j=2}^{n} \sum_{i=1}^{j-1} \left(\frac{1}{\Delta_{i,j}^2} \right) \Delta_{i,i+1},$$

up to time T.

Proof. We bound the regret of the algorithm arm by arm. Let us consider an arm $2 \leq j \leq n$. For $i < j$, let us count the number $N_{i,j}$ of times j was played when some arm in $1, 2, \ldots, i$ was awake. (In these iterations, the regret accumulated is at least $\Delta_{i,j}$ and at most $\Delta_{1,j}$.) We claim that $\mathbb{E}[N_{i,j}] \leq Q_{i,j}$, where $Q_{i,j} := \frac{33 \ln T}{\Delta_{i,j}^2}$.

We want to claim that after playing j for $Q_{i,j}$ number of times, we are unlikely to make the mistake of choosing j instead of something from the set $\{1, 2, \ldots, i\}$; that is, if the set of awake arms at time t includes some arm in $[i]$ as well as arm j, then with probability at least $1 - \frac{2}{t^4}$, some awake arm in $[i]$ will be chosen rather than arm j.

Let us bound the expected number of times j is chosen when $A(t) \cap [i] \neq \emptyset$ and j has already been played $Q_{i,j}$ number of times.

$$\begin{aligned}
&\sum_{Q_{i,j}<s\leq t\leq T} \mathbb{P}\Big[(x(t)=j) \wedge (j \text{ is played } s\text{-th time}) \wedge (A(t)\cap[i]\neq\emptyset)\Big] \\
&\leq \sum_{Q_{i,j}<s\leq t\leq T} \mathbb{P}\Big[(x(t)=j) \wedge (n_j(t)=s) \wedge \left(\forall_{k=1}^{i} \left(\hat{\mu}_j(t)+\rho_j(t) \geq \hat{\mu}_k(t)+\rho_k(t)\right)\right)\Big] \\
&= \sum_{Q_{i,j}<s\leq t\leq T} \mathbb{P}\Bigg[(x(t)=j) \wedge (n_j(t)=s) \\
&\qquad\qquad \wedge \left(\forall_{k=1}^{i} \left(\hat{\mu}_j(t)+\sqrt{\frac{8\ln t}{s}} \geq \hat{\mu}_k(t)+\rho_k(t)\right)\right)\Bigg] \\
&\leq \sum_{Q_{i,j}<s\leq t\leq T} \mathbb{P}\left[\forall_{k=1}^{i} \left(\hat{\mu}_j(t)+\sqrt{\frac{8\ln t}{s}} \geq \hat{\mu}_k(t)+\rho_k(t)\right)\right]. \qquad (4.2.10)
\end{aligned}$$

Let us define the event inside the probability expression as E_1 and define E_2 to be the event that $\hat{\mu}_k(t) \in [\mu_k - \rho_k(t), \mu_k + \rho_k(t)]$ for all $k \in \{j\} \cup \{1, 2, \ldots, i\}$. (Although E_1 and E_2 depend on s and t, we suppress this dependence for notational convenience.) The probability of event E_2 is at least $1 - (i+1)t^{-4}$ (from Lemma 4.2.9).

We will bound use the probability of E_1 by conditioning it on the event E_2. We can write $\mathbb{P}[E_1] = \mathbb{P}[E_1|E_2]\mathbb{P}[E_2] + \mathbb{P}[E_1|E_2^c]\mathbb{P}[E_2^c] \leq \mathbb{P}[E_1|E_2] + \mathbb{P}[E_2^c]$. To bound $\mathbb{P}[E_1|E_2]$, notice that the confidence $\rho_j(t)$ of arm j is at most $\sqrt{\frac{8\ln T}{33\ln T} \cdot \Delta_{i,j}^2} < \frac{\Delta_{i,j}}{2}$.

If event E_2 happens, $\hat{\mu}_j(t) + \rho_j(t) \leq (\mu_j + \rho_j(t)) + \rho_j(t) < \mu_j + \Delta_{i,j} = \mu_i$. Also, $\hat{\mu}_k(t) + \rho_k(t) \geq \mu_k$ for all $k = 1, 2, \ldots, i$. Therefore, the sum in (4.2.10) can be upper-bounded by following.

$$\begin{aligned}
&\sum_{Q_{i,j}<s\leq t\leq T} \left(\mathbb{P}\left[\forall_{k=1}^{i} \left(\hat{\mu}_j(t)+\sqrt{\frac{8\ln t}{s}} \geq \hat{\mu}_k(t)+\rho_k(t)\right) \,\middle|\, E_2\right] + \mathbb{P}[E_2^c]\right) \\
&\leq \sum_{Q_{i,j}<s\leq t\leq T} \left(\mathbb{P}\left[\forall_{k=1}^{i} (\hat{\mu}_j(t)+\rho_j(t) \geq \mu_k)\right]\right) + \sum_{Q_{i,j}<s\leq t\leq T} \frac{i+1}{t^4}
\end{aligned}$$

$$\leq \sum_{Q_{i,j}<s\leq t\leq T} \mathbb{P}\left[\hat{\mu}_j(t) + \rho_j(t) \geq \mu_i\right] + \sum_{Q_{i,j}<s\leq t\leq T} \frac{i+1}{t^4} \quad \text{(Since } \mu_1 \geq \mu_2 \geq \cdots \geq \mu_i.\text{)}$$

$$\leq \mathcal{O}(nT^{-2}) \quad \text{(The first term is zero, since } \hat{\mu}_j(t) + \rho_j(t) < \mu_i\text{, see above.)}$$

$$= \mathcal{O}(1).$$

Therefore, after j has been played $Q_{i,j}$ number of times, the expected number of additional times that j is played when $A(t) \cap [i] \neq \emptyset$ is bounded above by a constant. This implies

$$\mathbb{E}[N_{i,j}] \leq Q_{i,j} + \mathcal{O}(1) \leq \frac{33\ln(T)}{\Delta_{i,j}^2} + \mathcal{O}(1).$$

Now, it is easy to bound the total regret of the algorithm, which is

$$\mathbb{E}\left[\sum_{j=2}^{n}\sum_{i=1}^{j-1}(N_{i,j} - N_{i-1,j})\Delta_{i,j}\right] = \sum_{j=2}^{n}\sum_{i=1}^{j-1} N_{i,j}\left(\Delta_{i,j} - \Delta_{i+1,j}\right), \tag{4.2.11}$$

which follows by regrouping of terms and the convention that $N_{0,j} = 0$ and $\Delta_{j,j} = 0$ for all j. Taking the expectation of this gives the regret bound of

$$\left(33\ln T + \mathcal{O}(1)\right) \cdot \sum_{j=2}^{n}\sum_{i=1}^{j-1}\left(\frac{1}{\Delta_{i,j}^2}\right)(\Delta_{i,j} - \Delta_{i+1,j}).$$

This gives the statement of the lemma. □

Remarks for small $\Delta_{i,i+1}$ As noted in the case of the expert setting, the upper bound above becomes very weak if some $\Delta_{i,i+1}$ are small. In such a case, the proof can be modified by changing equation (4.2.11) as follows.

$$\sum_{j=2}^{n}\sum_{i=1}^{j-1}(N_{i,j} - N_{i-1,j})\Delta_{i,j}$$

$$= \sum_{j=2}^{n}\sum_{i=1}^{i_\epsilon(j)}(N_{i,j} - N_{i-1,j})\Delta_{i,j} \; + \; \sum_{j=2}^{n}\sum_{i=i_\epsilon(j)+1}^{j-1}(N_{i,j} - N_{i-1,j})\Delta_{i,j}$$

$$
\begin{aligned}
&\leq \sum_{j=2}^{n} \sum_{i=1}^{i_\epsilon(j)-1} N_{i,j}\Delta_{i,i+1} \;+\; \sum_{j=2}^{n} N_{i_\epsilon(j),j}\Delta_{i_\epsilon(j),j} \;+\; \sum_{j=2}^{n} \sum_{i=i_\epsilon(j)+1}^{j-1} (N_{i,j} - N_{i-1,j})\epsilon \\
&\leq \sum_{j=2}^{n} \sum_{i=1}^{i_\epsilon(j)-1} N_{i,j}\Delta_{i,i+1} \;+\; \epsilon \sum_{j=2}^{n} N_{i_\epsilon(j),j} \;+\; \epsilon \sum_{j=2}^{n} (N_{j-1,j} - N_{i_\epsilon(j),j}) \\
&\leq \sum_{1 \leq i < j \leq n, \Delta_{i,j} > \epsilon} N_{i,j}\Delta_{i,i+1} \;+\; \epsilon T,
\end{aligned}
$$

where the last step follows from $\sum_{j=2}^{n} N_{j-1,j} \leq T$.

Taking the expectation, and using the Lemma 4.2.5, we get the following regret bound for AUER algorithm.

Theorem 4.2.12. *For any $\epsilon \geq 0$, the regret of the* AUER *algorithm is at most*

$$
\epsilon T + \sum_{j=j_0(1)+1}^{n} \frac{33 \ln T + \mathcal{O}(1)}{\max\{\epsilon, \Delta_{i_0(j)-1,i_0(j)}\}} + \sum_{i=1}^{j_0(n)-1} \frac{33 \ln T + \mathcal{O}(1)}{\max\{\epsilon, \Delta_{j_0(i),j_0(i)+1}\}},
$$

up to time T.

Lower bound

In this section, we prove that the AUER algorithm presented is information theoretically optimal up to constant factors when the numbers μ_i — the mean payoffs of arms — are bounded away from 0 and 1. We do this by presenting a lower bound of

$$
\Omega\left(\ln T \cdot \sum_{i=1}^{n-1} \Delta_{i,i+1}^{-1}\right)
$$

for this problem. This is done by closely following the lower bound of Lai and Robbins (1985b) for two-armed bandit problems. The difference is that Lai and Robbins prove their lower bound only in the case when $T \to \infty$, but we want to get bounds that hold for finite T. Our main result is stated in the following lemma.

Lemma 4.2.13. *Let $P_i = \text{Bernoulli}(\mu_i)$ for $i = 1, 2, \ldots, n$ be payoff distributions with $\mu_i \in (\alpha, \beta)$ for some $0 < \alpha < \beta < 1$. Let ϕ be an algorithm for picking among*

n arms such that for all t, the expected number of times ϕ plays a suboptimal bandit up to time t is bounded above by $c_1 t^{0.1} + c_2$ (c_1 and c_2 possibly depend on μ_i). Then, there is an input instance with n arms endowed with some permutation of the aforementioned distributions $(P_i)_{i=1}^n$, such that the regret of ϕ is at least

$$\Omega\left(\sum_{i=1}^{n-1} \frac{(\log T)(\mu_i - \mu_{i+1})}{\mathsf{KL}(\mu_{i+1}; \mu_i)}\right),$$

for $T \geq T_0$, where T_0 is a function of n, μ_i, c_1, c_2, α, β.

We note that the exponent 0.1 in the lemma is quite arbitrary. Indeed, any nonzero exponent would work for the purpose of the proof.

Note that the above lower bound without the $(\log T)$ factor follows from the stochastic best expert lower bound in Lemma 4.2.7.

Using the fact that for $\mu_i \in (\alpha, \beta)$, $\mathsf{KL}(\mu_j; \mu_i) = \mathcal{O}_{\alpha,\beta}(\Delta_{i,j}^2)$, the lower bound can also be stated as

$$\Omega_{\alpha,\beta}\left(\sum_{i=1}^{n-1} \frac{(\log T)}{\Delta_{i,i+1}}\right),$$

which matches (up the constant factors) the upper bound in Theorem 4.2.10. Note that the notations $\mathcal{O}_{\alpha,\beta}(\cdot)$ and $\Omega_{\alpha,\beta}(\cdot)$ hide dependence on α and β.

We first prove the result for two arms. For this, in the following, we extend the Lai and Robbins result so that it holds (with somewhat worse constants) for finite T, rather than only in the limit $T \to \infty$.

Lemma 4.2.14. *Let $P_i = \text{Bernoulli}(\mu_i)$ for $i = 1, 2$ with $\mu_2 < \mu_1$, $\mu_i \in (\alpha, \beta)$ for $i = 1, 2$ and $0 < \alpha < \beta < 1$. Let ϕ be any algorithm for choosing among two arms which never picks the worse arm (for any values of μ_1 and μ_2 in (α, β)) more than $c_1 t^{0.1} + c_2$ times up to time t (c_1 and c_2 possibly depend on μ_1 and μ_2). Then there*

exists an instance with two arms endowed with two distributions above (in some order) such that the regret of the algorithm ϕ when presented with this instance is at least

$$\frac{1}{6}\left(\frac{(\log T)(\mu_1-\mu_2)}{\mathsf{KL}(\mu_2;\mu_1)}\right),$$

for all $T \geq T_0$, and the value of T_0 can be explicitly computed as a function of μ_1, μ_2, c_1, c_2, α, β.

Proof. From the assumption that μ_1 and μ_2 are bounded away from 0 and 1, there exists a Bernoulli distribution with mean $\lambda > \mu_1$ with

$$|\,\mathsf{KL}(\mu_2;\lambda) - \mathsf{KL}(\mu_2;\mu_1)| \leq \frac{1}{10}\cdot \mathsf{KL}(\mu_2;\mu_1),$$

because of the continuity of KL divergence in its second argument. Indeed, using the convexity of $\mathsf{KL}(\mu_2;\cdot)$ (for fixed μ_2), and the fact that the slope of $\mathsf{KL}(\mu_2;\cdot)$ is bounded by $\frac{\beta-\mu_2}{\beta(1-\beta)}$, λ can be chosen to be $\min\left\{\mu_1 + \frac{\mathsf{KL}(\mu_2;\mu_1)}{10}\frac{\beta(1-\beta)}{(\beta-\mu_2)}, \frac{\beta-\mu_1}{2}\right\}$. This claim provides us with a Bernoulli distribution with mean λ (which is an explicit function of μ_i and β) such that

$$\mathsf{KL}(\mu_2;\lambda) \leq \frac{11}{10}\cdot \mathsf{KL}(\mu_2;\mu_1). \tag{4.2.12}$$

From now on, until the end of the proof, we work with the following two distributions on T-step histories: p is the distribution induced by the algorithm ϕ playing against Bernoulli arms with means (μ_1, μ_2), and q is the distribution induced by ϕ playing against Bernoulli arms with means (μ_1, λ). From the assumption of the lemma, we have

$$\mathbb{E}_q[T - n_2(T)] \leq c_1 T^{0.1} + c_2.$$

Note that c_1 and c_2 here are functions of μ_1 and λ (which in turn is a function of μ_i, α, β). By an application of Markov's inequality, we get that

$$\mathbb{P}_q\left[n_2(T) < \frac{9}{10}(\log T)/\,\mathsf{KL}(\mu_2;\lambda)\right] \leq \frac{\mathbb{E}_q[T-n_2(T)]}{T - \frac{9}{10}(\log T)/\,\mathsf{KL}(\mu_2;\lambda)}$$

$$
\begin{aligned}
&\leq \frac{c_1 T^{0.1} + c_2}{T/2} \qquad \left(\text{for } T > e^{5/(9\,\mathsf{KL}(\mu_2;\lambda))}\right) \\
&\leq 4c_1 T^{-0.9}. \qquad (\text{for } T > (c_2/c_1)^{10})
\end{aligned} \tag{4.2.13}
$$

Let $\mathcal{E}$ denote the event that $n_2(T) < \frac{9}{10}(\log T)/\,\mathsf{KL}(\mu_2;\lambda)$. If $\mathbb{P}_p(\mathcal{E}) < 1/3$, then

$$
\begin{aligned}
\mathbb{E}_p[n_2(T)] &\geq \mathbb{P}_p(\overline{\mathcal{E}}) \cdot \frac{9}{10}\,(\log T)\,/\,\mathsf{KL}(\mu_2,\lambda) \\
&\geq \frac{2}{3} \cdot \frac{9}{10} \cdot \frac{\log T}{\mathsf{KL}(\mu_2,\lambda)} \\
&\geq \frac{2}{3} \cdot \frac{9}{11} \cdot \frac{\log T}{\mathsf{KL}(\mu_2;\mu_1)},
\end{aligned}
$$

which implies the stated lower bound.

Henceforth, we will assume $\mathbb{P}_p(\mathcal{E}) \geq 1/3$. We have $\mathbb{P}_q(\mathcal{E}) < 1/3$ using (4.2.13). Now applying the lemma from Karp and Kleinberg (2007b) stated in (4.2.9), we have

$$
\begin{aligned}
\mathsf{KL}(p;q) &\geq \frac{1}{3}\ln\left(\frac{1}{3 \cdot 4c_1 T^{-0.9}}\right) - \frac{1}{e} \\
&= \frac{1}{3}(0.9)\ln T - \left(\frac{1}{e} + \frac{1}{3}\ln(12c_1)\right) \\
&= (0.3)\ln T - \left(\frac{1}{3}\ln(e^{3/e}c_1)\right).
\end{aligned} \tag{4.2.14}
$$

The chain rule for KL divergence (Cover and Thomas, 1999, Theorem 2.5.3) implies

$$
\mathsf{KL}(p;q) = \mathbb{E}_p[n_2(T)] \cdot \mathsf{KL}(\mu_2;\lambda) \tag{4.2.15}
$$

Combining (4.2.14) with (4.2.15), we get

$$
\begin{aligned}
\mathbb{E}_p[n_2(T)] &\geq \frac{(0.3)\ln T - \frac{1}{3}\ln(e^{3/e}c_1)}{\mathsf{KL}(\mu_2;\lambda)} \\
&\geq \frac{0.3}{1.1}\frac{\ln T}{\mathsf{KL}(\mu_2;\mu_1)} - \frac{1}{3}\frac{\ln(e^{3/e}c_1)}{\mathsf{KL}(\mu_2;\mu_1)} \\
&= \frac{3}{11}\frac{\ln(T)}{\mathsf{KL}(\mu_2;\mu_1)} - \frac{1}{3}\frac{\ln(e^{3/e}c_1)}{\mathsf{KL}(\mu_2;\mu_1)}
\end{aligned}
$$

$$\geq \frac{3}{11}\frac{\ln(T)}{\mathsf{KL}(\mu_2;\mu_1)} - \frac{1}{10}\frac{\ln(T)}{\mathsf{KL}(\mu_2;\mu_1)} \quad (\text{for } T > (e^{3/e}c_1)^{10/3})$$
$$\geq \frac{1}{6}\frac{\ln(T)}{\mathsf{KL}(\mu_2;\mu_1)}.$$

This gives the required regret bound. The explicit value of T above which the bound holds is

$$T_0 := \max\left\{e^{5/(9\,\mathsf{KL}(\mu_2;\lambda))},\ \left(\frac{c_2}{c_1}\right)^{10},\ (e^{3/e}c_1)^{10/3}\right\},$$

which can be explicitly written as a function of μ_1, μ_2, c_1, c_2, α, β. □

We now extend the result from 2 to n bandits.

Proof of Lemma 4.2.13: A naive way to extend the lower bound is to divide the time line between $n/2$ blocks of length $2T/n$ each and use $n/2$ separate two-armed bandit lower bounds as done in the proof of Lemma 4.2.7.

We can pair the arms in pairs of $(2i-1, 2i)$ for $i = 1, 2, \ldots, \lfloor n/2 \rfloor$. We present the algorithm with two arms $2i-1$ and $2i$ in the i-th block of time. The lower bound then is

$$\Omega\left(\log\left(\frac{T}{n}\right)\sum_{i=1}^{\lfloor n/2\rfloor}\left(\frac{\mu_{2i-1}-\mu_{2i}}{\mathsf{KL}(\mu_{2i};\mu_{2i-1})}\right)\right).$$

We get a similar lower bound by presenting the algorithm with $(2i, 2i+1)$:

$$\Omega\left(\log\left(\frac{T}{n}\right)\sum_{i=1}^{\lfloor (n-1)/2\rfloor}\left(\frac{\mu_{2i}-\mu_{2i+1}}{\mathsf{KL}(\mu_{2i+1};\mu_{2i})}\right)\right).$$

Taking the average of the two lower bounds and $T \geq n^2$ gives the required lower bound of

$$\Omega\left(\sum_{i=1}^{n-1}\frac{(\log T)(\mu_i-\mu_{i+1})}{\mathsf{KL}(\mu_{i+1};\mu_i)}\right),$$

finishing the proof of the lemma. □

4.3 Adversarial Model of Rewards

We now turn our attention to the case where no distributional assumptions are made on the generation of rewards. We consider in turn the best expert setting and the multi-armed bandit setting. For each setting, we first prove information theoretic lower bounds on the regret of any online learning algorithm, and then present online algorithms whose regret is within a constant factor of the lower bound for the expert setting and within a sub-logarithmic factor of the lower bound for the bandit setting. Unlike in the stochastic rewards setting, however, these algorithms are not computationally efficient. It is an open problem if there exists an efficient algorithm whose regret grows as $\mathcal{O}(T^{1-\epsilon}n^c)$ for some positive constants ϵ, c.

4.3.1 Best Expert Setting

In this section, we consider the adversarial sleeping best expert setting. Recall that in the sleeping best expert setting, the algorithm chooses an expert to play in each time round from the set of available experts, and at the end of the round, gets to observe the rewards of *all* available experts for that round, not just for the one it chose. There is no assumption on how the rewards of these experts are generated in each round; indeed an adversary chooses the reward of each expert in each time round, and can observe the choices made by the algorithm prior to that round in choosing the rewards for a particular round. Additionally, the adversary also chooses which subset of the experts will be awake (available) in each time round.

We first present a lower bound on the achievable regret of any algorithm for the adversarial sleeping best expert problem.

Theorem 4.3.1. *For every online algorithm* ALG *and every time horizon T, there is an adversary such that the algorithm's regret with respect to the best ordering, at time T, is*

$$\Omega(\sqrt{Tn\log(n)}).$$

Proof. We construct a randomized oblivious adversary (i.e. a distribution on input sequences of length T) such that the regret of any algorithm ALG is at least $\Omega(\sqrt{Tn\log(n)})$. The adversary partitions the timeline $\{1, 2, \ldots, T\}$ into a series of *two-expert games*, i.e. intervals of consecutive rounds during which only two experts are awake and all the rest are asleep. In total there will be $Q(n) = \Theta(n \log n)$ two-expert games, where $Q(n)$ is a function to be specified later in (4.3.2). For $i = 1, 2, \ldots, Q(n)$, the set of awake experts throughout the i-th two-experts game is a pair $A^{(i)} = \{x_i, y_i\}$, determined by the adversary based on the (random) outcomes of previous two-experts games. The precise rule for determining the elements of $A^{(i)}$ will be explained later in the proof.

Each two-experts game runs for $T_0 = T/Q(n)$ rounds, and the payoff functions for the rounds are independent, random bijections from $A^{(i)}$ to $\{0, 1\}$. Letting $g^{(i)}(x_i)$, $g^{(i)}(y_i)$ denote the total payoffs of x_i and y_i, respectively, during the two-experts game, it follows from Khintchine's inequality (Khintchine, 1923) that

$$\mathbb{E}\left(\left|g^{(i)}(x_i) - g^{(i)}(y_i)\right|\right) = \Omega\left(\sqrt{T_0}\right). \tag{4.3.1}$$

The expected payoff for any algorithm can be at most $\frac{T_0}{2}$, so for each two-experts game the regret of any algorithm is at least $\Omega(\sqrt{T_0})$. For each two-experts game we define the *winner* W_i to be the element of $\{x_i, y_i\}$ with the higher payoff in the two-experts game; we will adopt the convention that $W_i = x_i$ in case of a tie. The *loser* L_i is the element of $\{x_i, y_i\}$ which is not the winner.

The adversary recursively constructs a sequence of $Q(n)$ two-experts games and an ordering of the experts such that the winner of every two-experts game precedes the loser in this ordering. (We call such an ordering *consistent* with the sequence of games.) In describing the construction, we assume for convenience that n is a power of 2. If $n = 2$ then we set $Q(2) = 1$ and we have a single two-experts game and an ordering in which the winner precedes the loser. If $n > 2$ then we recursively construct a sequence of games and an ordering consistent with those games, as follows:

1. We construct $Q(n/2)$ games among the experts in the set $\{1, 2, \ldots, n/2\}$ and an ordering $\prec_1$ consistent with those games.
2. We construct $Q(n/2)$ games among the experts in the set $\{(n/2)+1, \ldots, n\}$ and an ordering $\prec_2$ consistent with those games.
3. Let $k = 2Q(n/2)$. For $i = 1, 2, \ldots, n/2$, we define x_{k+i} and y_{k+i} to be the i-th elements in the orderings $\prec_1, \prec_2$, respectively. The $(k+i)$-th two-experts game uses the set $A^{(k+i)} = \{x_{k+i}, y_{k+i}\}$.
4. The ordering of the experts puts the winner of the game between x_{k+i} and y_{k+i} before the loser, for every $i = 1, 2, \ldots, n/2$, and it puts both elements of $A^{(k+i)}$ before both elements of $A^{(k+i+1)}$.

By construction, it is clear that the ordering of experts is consistent with the games, and that the number of games satisfies the recurrence

$$Q(n) = 2Q(n/2) + n/2, \tag{4.3.2}$$

whose solution is $Q(n) = \Theta(n \log n)$.

The best ordering of experts achieves a payoff at least as high as that achieved by the constructed ordering which is consistent with the games. By (4.3.1), the

expected payoff of that ordering is $T/2 + Q(n) \cdot \Omega(\sqrt{T_0})$. The expected payoff of ALG in each round t is $1/2$, because the outcome of that round is independent of the outcomes of all prior rounds. Hence the expected payoff of ALG is only $T/2$, and its regret is

$$Q(n) \cdot \Omega(\sqrt{T_0}) = \Omega(n \log n \sqrt{T/(n \log n)}) = \Omega(\sqrt{Tn \log n}).$$

This proves the theorem. □

It is interesting to note that the adversary that achieves this lower bound is not adaptive in either choosing the payoffs or choosing the awake experts at each time step, i.e. it makes these choices without considering the algorithm's past decisions. It only needs to be able to carefully coordinate which experts are awake based on the payoffs at previous time steps.

Even more interesting is the fact that this lower bound is tight, so an adaptive adversary is not more powerful than an oblivious one. There is a learning algorithm that achieves a regret of $\mathcal{O}(\sqrt{Tn \log(n)})$. We turn our attention to this algorithm now.

To achieve this regret we transform the sleeping experts problem to a problem with $n!$ experts that are always awake, and we choose among these $n!$ experts using the Hedge algorithm (see (Freund and Schapire, 1999), and Section 2.1). In the transformed problem, we have one expert for each σ-policy (i.e. ordering of the original n experts). At each round, each of the $n!$ experts makes a prediction according to its corresponding σ-policy, (i.e. the same prediction as the highest ranked awake expert in the corresponding ordering), and receives the payoff of that policy (i.e. the payoff of the highest ranked awake expert in the corresponding ordering).

Theorem 4.3.2. *An algorithm that makes predictions using the* Hedge *algorithm on the transformed problem achieves* $\mathcal{O}(\sqrt{Tn\log(n)})$ *regret with respect to the best ordering.*

Proof. Every expert in the transformed problem receives the payoff of its corresponding ordering in the original problem. Since Hedge achieves regret $\mathcal{O}(\sqrt{T\log(n!)})$ with respect to the best expert in the transformed problem, the same regret is achieved by the algorithm in the original problem. The theorem follows by applying the bound $\log(n!) = \mathcal{O}(n\log n)$, which is a consequence of Stirling's formula. □

In a naive implementation the algorithm described above is obviously not computationally efficient since in each round we have to sample among $n!$ experts and update $n!$ weights. A natural question is whether this algorithm can be implemented in polynomial time by devising an efficient sampling scheme and a clever weight update procedure. The following theorem, unfortunately, shatters any hope that this might be possible.

Theorem 4.3.3. *Unless* RP = NP*, any learning algorithm for the adversarial sleeping experts problem that:*

1. *generates its output by sampling over σ-policies, independently of the set of awake experts*
2. *has regret bounded by $T^{1-\epsilon}\cdot p(n)$ for some $\epsilon > 0$ and some polynomial function $p(\cdot)$*

cannot be implemented in polynomial time.

Proof. We prove this theorem via a reduction from the minimum feedback arc set problem (Garey and Johnson, 1979). The notion of reduction here is not the usual Karp-reduction, but we will show that if there is an algorithm with specified conditions, then we can find the optimum for any feedback arc set instance with probability at least $1 - \delta$ for any constant $\delta > 0$.

Let ALG be any algorithm that respects the conditions in the theorem. We are given a directed graph $G = (V, A)$, in which we are to find the minimum feedback arc set. Every permutation of the vertices defines a feedback arc set, but this mapping is not one to one. (There can be many permutations for one feedback arc set.) For a permutation σ, the corresponding feedback arc set is the set of arcs going from higher numbered vertices to lower numbered vertices, i.e. , $\{a = (u, v) \in A : \sigma(u) > \sigma(v)\}$. The cardinality of this set is denoted by $\texttt{FAS}(\sigma)$. For a feedback arc set $A' \subseteq A$, a corresponding permutation can be found by choosing one of the topological orderings of the graph $(G, A \setminus A')$. It is easy to see that the minimum feedback arc set is equal to $\min_\sigma \texttt{FAS}(\sigma)$. We will use the learning algorithm ALG to find, with high probability, an ordering σ minimizing $\texttt{FAS}(\sigma)$.

We instantiate an adversarial sleeping experts problem with $|V|$ experts, one for each vertex in the graph. In each round, the adversary selects an arc (u, v) in A uniformly at random and makes the two experts corresponding to the head (v) and the tail (u) of the selected arc awake and all the other experts asleep. It then associates a payoff of 1 to the expert corresponding to the tail of the arc and a payoff of 0 to the expert corresponding to the head of the arc. We play for $T := 2(\lceil \frac{1}{\delta} \rceil p(n)m)^{1/\epsilon}$ rounds and in each round we record the σ-policy selected by ALG and also the feedback arc set value of the permutation σ. At the end of the

1 Let $\sigma = (1, 2, \ldots, |V|)$ (current best permutation) and $x = \texttt{FAS}(\sigma)$ (value of the best feedback arc set so far).

2 **FOR** $t = 1$ to $T = 2(\lceil \frac{1}{\delta} \rceil p(n)m)^{1/\epsilon}$

3 Choose $(u, v) \in A$ at random from m arcs in A. Let $\{u, v\}$ be the set of awake experts. Set the payoff of u to 1 and the payoff of v to 0.

4 Record the permutation σ_t that the algorithm ALG outputs.

5 **IF** $x > \texttt{FAS}(\sigma_t)$

6 $\sigma \leftarrow \sigma_t$

7 $x \leftarrow \texttt{FAS}(\sigma_t)$

8 Output $\{a = (u, v) \in A : \sigma(u) > \sigma(v)\}$ as the feedback arc set.

Figure 4.3: Algorithm to solve Feedback Arc Set Problem from low regret adversarial expert algorithm.

T rounds we choose the best permutation among the T rounds — the one with the smallest $\texttt{FAS}(\sigma)$ value — and output the corresponding feedback arc set. See Algorithm 4.3.

Let $\texttt{FAS}_*$ be the optimum value of the feedback arc set. Let σ be the permutation selected by Algorithm 4.3. We claim that $\texttt{FAS}(\sigma) = \texttt{FAS}_*$ with probability at least $1 - \delta$. Since the number of rounds is polynomial in n and m (ϵ and δ are constants), this will solve feedback arc set in randomized polynomial time.

Since the expected regret of the algorithm is at most $T^{1-\epsilon}p(n)$, it follows from Markov's inequality that with probability at least $1 - \delta$, the regret is at most $\frac{1}{\delta}T^{1-\epsilon}p(n)$. We will prove that in this event, our algorithm finds a σ with $\texttt{FAS}(\sigma) = \texttt{FAS}_*$.

We prove this claim by contradiction. If not, then for all $t = 1, 2, \ldots, T$,

$\texttt{FAS}(\sigma_t) \geq \texttt{FAS}_* + 1$. The expected reward of choosing a permutation τ is $1 - \frac{\texttt{FAS}(\tau)}{m}$. Therefore the expected regret in each round is at least

$$\left(1 - \frac{\texttt{FAS}_*}{m}\right) - \left(1 - \frac{\texttt{FAS}(\sigma_t)}{m}\right) \geq \frac{1}{m}.$$

Hence, the total regret of the algorithm is at least $T \cdot \frac{1}{m}$. We also know that the regret is at most $\frac{1}{\delta} T^{1-\epsilon} p(n)$. This gives the following relation:

$$\frac{T}{m} \leq \frac{1}{\delta} T^{1-\epsilon} p(n),$$

which simplifies to $T \leq (\frac{1}{\delta} p(n) m)^{1/\epsilon}$, a contradiction since we have taken T to be twice as much. This proves that if we run our algorithm for $T := 2(\lceil \frac{1}{\delta} \rceil p(n) m)^{1/\epsilon}$, then with probability at least $1 - \delta$, we recover the optimum feedback arc set for the graph. This proves the theorem. □

Note that this does not mean that there does not exist an efficient, low regret algorithm for the adversarial sleeping experts problem. One might be able to design an efficient, low regret algorithm that either does not sample over σ-policies, or makes the sampling dependent on the set of awake experts. For instance, there exists a simple algorithm that achieves low regret against the particular adversary used in the proof above: run a separate instance of the Hedge algorithm for every pair of experts and, in each round, use the instance of Hedge corresponding to the two experts that are awake. Since the adversary will only present the algorithm with two awake experts at a time, this algorithm can always make a prediction, and its regret will be bounded by $\mathcal{O}(\sqrt{T \cdot n^2})$.

4.3.2 Multi-Armed Bandit Setting

Finally, we consider the adversarial sleeping multi-armed bandit setting. Recall that in the sleeping multi-armed bandit setting, the algorithm chooses an arm to

play in each round from the set of available arms, and at the end of the round, gets to observe the rewards of the *chosen* arm (unlike best expert setting, where the algorithm observes the reward of all potential choices). There is no assumption on how the rewards of these arms are generated in each round. Additionally, an adversary also chooses which subset of arms will be awake (available to be chosen by the algorithm) in each round.

We first present a lower bound on the achievable regret of any algorithm for the adversarial sleeping multi-armed bandit problem.

Theorem 4.3.4. *For every online algorithm* ALG *and every time horizon* T*, there is an adversary such that the algorithm's regret with respect to the best ordering, at time* T*, is* $\Omega(n\sqrt{T})$.

Proof. To prove the lower bound we will rely on the lower bound proof for the standard multi-armed bandit when all the bandits are awake (Auer et al., 2002a). In the standard "all-awake" bandit setting with a time horizon of T_0, any algorithm will have at least $\Omega(\sqrt{T_0 n})$ regret with respect to the best bandit. To ensure this regret, the input sequence is generated by sampling T_0 times independently from a distribution in which every bandit but one receives a payoff of 1 with probability $\frac{1}{2}$ and 0 otherwise. The remaining bandit, which is chosen at random, incurs a payoff of 1 with probability $\frac{1}{2} + \epsilon$ for an appropriate choice of ϵ.

To obtain the lower bound for the sleeping bandits setting we set up a sequence of n multi-armed bandit games as described above. Each game will run for $T_0 = \frac{T}{n}$ rounds. The bandit that received the highest payoff during the game will become asleep and unavailable in the rest of the games.

In game i, any algorithm will have a regret of at least $\Omega\left(\sqrt{\frac{T}{n}(n-i)}\right)$ with

respect to the best bandit in that game. Consequently, the regret of any learning algorithm with respect to the best ordering is bounded below by a positive constant times the following expression:

$$\sum_{i=1}^{n-1} \sqrt{\frac{T}{n}(n-i)} = \sqrt{\frac{T}{n}} \sum_{j=1}^{n-1} j^{1/2} \geq \sqrt{\frac{T}{n}} \int_{x=0}^{n-1} x^{1/2} dx$$
$$= \frac{2}{3}\sqrt{\frac{T}{n}} \cdot (n-1)^{3/2} = \Omega\left(n\sqrt{T}\right).$$

The theorem follows. □

Let us now turn our attention to getting an algorithm for the adversarial sleeping multi-armed bandit problem. To get an upper bound on regret, we will use the Exp4 algorithm (see Section 2.4.1, and (Auer et al., 2002a)).

Exp4 chooses an arm by combining the advice of a set of "experts". At each round, each expert provides advice in the form of a probability distribution over arms. In particular the advice can be a point distribution concentrated on a single action. (It is required that at least one of the experts is the *uniform expert* whose advice is always the uniform distribution over arms.)

To use Exp4 for the sleeping experts setting, we concoct $n! + 1$ "experts", one corresponding to the "uniform" expert which chooses each arm with equal probability, and one each for $n!$ orderings. The expert corresponding to an ordering σ always "advises" to play the arm $\texttt{first}(A(t), \sigma)$ (first available arm in its ordering), i.e., in each round, the advice of expert σ is a point distribution concentrated on the highest ranked arm in the corresponding ordering σ.

This introduces a slight problem. Since the uniform expert may advise us to

pick arms which are not awake, we assume for convenience that the algorithms is *not* restricted to choose an action from $A(t)$ (awake set), but is allowed to choose any action at all, with the proviso that the payoff of an action in the complement of $A(t)$ is defined to be 0. Note that any algorithm for this modified problem can easily be transformed into an algorithm for the original problem: every time the algorithm chooses an action in the complement of $A(t)$ we instead play an arbitrary action in $A(t)$ (and don't use the feedback obtained about its payoff). Such a transformation can only increase the algorithm's payoff, i.e. decrease the regret. Hence, to prove the regret bound asserted in Theorem 4.3.5 below, it suffices to prove the same regret bound for the case when algorithm is allowed to choose an arm in complement of $A(t)$ with zero payoff.

Theorem 4.3.5. *The* Exp4 *algorithm as described above achieves a regret of* $\mathcal{O}(n\sqrt{T\log(n)})$ *with respect to the best ordering, against an adaptive adversary.*

Proof. We have n arms and $1+n!$ experts, so the regret of Exp4 with respect to the payoff of the best expert is $\mathcal{O}(\sqrt{Tn\log(n!+1)})$ (Auer et al., 2002a). Using the estimate $\log(n!+1) = \mathcal{O}(n\log n)$, this regret bound can be rewritten as $\mathcal{O}(n\sqrt{T\log n})$. Since the payoff of each expert is exactly the payoff of its corresponding ordering, we obtain the statement of the theorem. □

The upper bound and lower bound differ by a factor of $\mathcal{O}(\sqrt{\log(n)})$, the gap resulting from adapting the Exp4 algorithm to our setting. In the classical multi-armed bandit setting, Audibert and Bubeck (2009) closed a similar gap ($\mathcal{O}(\sqrt{Tn\log n})$ upper bound versus $\Omega(\sqrt{Tn})$ lower bound) by improving the Exp3 algorithm. It is not clear how the policies from Audibert and Bubeck (2009) can be adapted for Exp4 algorithm, so closing the $\mathcal{O}(\sqrt{\log n})$ gap in the sleeping multi-armed bandit problem setting remains an important open problem.

CHAPTER 5

DISCUSSION AND CONCLUSIONS

The potential of applying online learning framework in variety of diverse scenarios combined with some crucial features of applications missing from the framework lead us to work on extending it in two interesting directions.

In Chapter 3, we considered an extension which allows for arms to be strategic agents who are trying to maximize their own utility. We are able to rigorously prove a performance separation (in terms of regret) between algorithms for the multi-armed bandit problem and truthful mechanisms for the multi-armed bandit mechanism design problem (which can be viewed as a strategic analogue of multi-armed bandit problem) with respect to implementation in dominant strategies (it is dominant strategy for agents to tell the truth, irrespective of clicks and other agents' bids). This leads to natural questions about how far we can push this separation, in terms of relaxing the notion of truthfulness from dominant strategies to something weaker. Are there solution concepts in which this performance separation is reduced (or disappears completely)? Can we prove structural results for these solution concepts?

Weakly truthful randomized mechanism A randomized mechanism $(\mathcal{A}, \mathcal{P})$ is called weakly truthful if for every realization of the clicks, the expected utility of agent i by bidding the true value is at least as much as expected utility by bidding anything else (the expectation is taken over algorithm's random seed).

It remains an open question to formulate and prove results about the structure of weakly truthful mechanisms, and if these mechanisms suffer more regret than best multi-armed bandit algorithms.

Truthfulness in expectation A mechanism $(\mathcal{A}, \mathcal{P})$ is said to be truthful in expectation if the expected utility of agent i by bidding her true value is at least as much as expected utility from bidding anything else (the expectation is now taken over random seed of the algorithm *and* clicks). In (Babaioff et al., 2010), it is proved that any "monotone" algorithm for the multi-armed bandit problem can be turned into a randomized mechanism for the truthful multi-armed bandit problem that is truthful-in-expectation and that is *not* normalized. Therefore, non-normalized, truthful-in-expectation mechanisms can achieve as good regret as achieved by best multi-armed bandit algorithms.

It is an interesting open question if *normalized* and *truthful-in-expectation* mechanism can perform as well as best multi-armed bandit algorithms. Note that mechanism derived in Section 3.8.4 is not normalized.

In Chapter 4, we have analyzed algorithms for full-information and partial-information prediction problems in the "sleeping experts" setting, using a novel benchmark which compares the algorithm's payoff against the best payoff obtainable by selecting available actions using a fixed total ordering of the actions. We have presented algorithms whose regret is (almost) information-theoretically optimal in both the stochastic and adversarial cases.

Computationally efficient algorithms In the stochastic case, our algorithms for sleeping expert and bandit problems are simple and computationally efficient. In the adversarial case, the most important open question is whether there is a computationally efficient algorithm which matches (or nearly matches) the regret bounds achieved by the exponential-time algorithms presented here. Or even before that, are their computationally efficient algorithms that achieve sublinear regret (not necessarily information-theoretically optimal)? The result in Theo-

rem 4.3.3 points to some difficulty in finding such algorithm, but still leaves open the possibility of an efficient algorithm.

Closing the logarithmic gap In the adversarial multi-armed bandit setting, we saw that the best algorithm we can find has regret bounded by $\mathcal{O}(\sqrt{Tn^2 \log n})$, while the lower bound places a bound of $\Omega(\sqrt{Tn^2})$ on the regret, leaving an $\mathcal{O}(\sqrt{\log n})$ gap. A similar gap in the all-awake multi-armed bandit setting was recently bridged by Audibert and Bubeck (2009). It is an interesting open question to determine if similar techniques can be used to bridge the gap in sleeping multi-armed bandit settings.

BIBLIOGRAPHY

Abernethy, J., E. Hazan, and A. Rakhlin. 2008. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Conf. on Learning Theory (COLT)*, 263–274.

Aggarwal, G., A. Goel, and R. Motwani. 2006. Truthful auctions for pricing search keywords. In *ACM Conf. on Electronic Commerce (EC)*, 1–7.

Aggarwal, G., and S. Muthukrishnan. 2008. Tutorial on theory of sponsored search auctions. In *IEEE Symp. on Foundations of Computer Science (FOCS)*.

Archer, A., and É. Tardos. 2001. Truthful mechanisms for one-parameter agents. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 482–491.

Athey, S., and I. Segal. 2007, March. An efficient dynamic mechanism. Available from http://www.stanford.edu/~isegal/agv.pdf.

Audibert, J.-Y., and S. Bubeck. 2009. Minimax policies for adversarial and stochastic bandits. In *COLT*.

Auer, P., N. Cesa-Bianchi, and P. Fischer. 2002a. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47 (2-3): 235–256.

Auer, P., N. Cesa-Bianchi, and P. Fischer. 2002b. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47 (2–3): 235–256. Preliminary version in *15th ICML*, 1998.

Auer, P., N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. 2002a. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.* 32 (1): 48–77.

Auer, P., N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. 2002b. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.* 32 (1): 48–77. Preliminary version in *36th IEEE FOCS*, 1995.

Awerbuch, B., and R. Kleinberg. 2008, February. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences* 74 (1): 97–114. Preliminary version appeared in STOC 2004.

Azuma, K. 1967. Weighted sums of certain dependent random variables. *Tohoku Math. J.* 19:357–367.

Babaioff, M., R. Kleinberg, and A. Slivkins. 2010. Truthful mechanisms with implicit payment computation. In *11th ACM Conf. on Electronic Commerce (EC)*.

Babaioff, M., Y. Sharma, and A. Slivkins. 2009. Characterizing truthful multiarmed bandit mechanisms. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC)*, 79–88.

Bartlett, P. L., V. Dani, T. Hayes, S. Kakade, A. Rakhlin, and A. Tewari. 2008. High-probability regret bounds for bandit online linear optimization. In *Conf. on Learning Theory (COLT)*, 335–342.

Ben-Or, M., and A. Hassidim. 2008. The Bayesian Learner is Optimal for Noisy Binary Search (and Pretty Good for Quantum as Well). In *IEEE Symp. on Foundations of Computer Science (FOCS)*.

Bergemann, D., and J. Välimäki. 2006, October. Efficient dynamic auctions. Available from cowles.econ.yale.edu/P/cd/d15b/d1584.pdf.

Berry, D., and B. Fristedt. 1985. *Bandit problems: sequential allocation of experiments*. Chapman&Hall.

Berry, D. A., and L. M. Pearson. 1985. Optimal designs for two-stage clinical trials with dichotomous responses. *Statistics in Medicine* 4:487–508.

Blum, A., and Y. Mansour. 2005. From external to internal regret. In *COLT*, 621–636.

Cesa-Bianchi, N., Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. 1997. How to use expert advice. *J. ACM* 44 (3): 427–485.

Cesa-Bianchi, N., and G. Lugosi. 2006. *Prediction, learning, and games*. Cambridge University Press.

Cover, T. M., and J. A. Thomas. 1991. *Elements of Information Theory*. New York: John Wiley & Sons.

Cover, T. M., and J. A. Thomas. 1999. *Elements of information theory*. J. Wiley.

Dani, V., and T. P. Hayes. 2006. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 937–943.

Devanur, N., and S. M. Kakade. 2009. The price of truthfulness for pay-per-click auctions. In *10th ACM Conf. on Electronic Commerce (EC)*, 99–106.

Dobzinski, S., and M. Sundararajan. 2008. On characterizations of truthful mechanisms for combinatorial auctions and scheduling. In *ACM Conf. on Electronic Commerce (EC)*, 38–47.

Edelman, B., M. Ostrovsky, and M. Schwarz. 2007, March. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review* 97 (1): 242–259.

Freund, Y., and R. E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* 55 (1): 119–139.

Freund, Y., and R. E. Schapire. 1999. Adaptive game playing using multiplicative weights. *Games and Economic Behavior* 29:79–103.

Freund, Y., R. E. Schapire, Y. Singer, and M. K. Warmuth. 1997. Using and combining predictors that specialize. In *STOC*, 334–343.

Garey, M. R., and D. S. Johnson. 1979. *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company.

Gittins, J. C. 1979. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B* 41 (2): 148–177.

Gittins, J. C. 1989. *Multi-Armed Bandit Allocation Indices*. John Wiley & Sons.

Gittins, J. C., and D. M. Jones. 1979. A dynamic allocation index for the discounted multiarmed bandit problem. *Biometrika* 66 (3): 561–565.

Gonen, R., and E. Pavlov. 2007. An incentive-compatible multi-armed bandit mechanism. In *ACM Symp. on Principles Of Distributed Computing (PODC) (Brief Announcement)*, 362–363. Preliminary version in *3rd Workshop on Sponsored Search Auctions* (in conjunction with *WWW* 2007).

Hannan, J. 1957. Approximation to Bayes risk in repeated plays. In *M. Dresher, A. Tucker, P. Wolfe (Eds.), Contributions to the Theory of Games, Princeton University Press*, Volume 3, 97–139.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *J. American Stat. Assoc.* 58:13–30.

Immorlica, N., K. Jain, M. Mahdian, and K. Talwar. 2005. Click fraud resistant methods for learning click-through rates. In *Intl. Workshop On Internet And Network Economics (WINE)*, 34–45.

Kalai, A. T., and S. Vempala. 2005. Efficient algorithms for on-line optimization. *J. Computer and System Sciences* 71 (3): 291–307.

Karp, R., and R. Kleinberg. 2007a. Noisy binary search and its applications. In *18th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 881–890.

Karp, R. M., and R. Kleinberg. 2007b. Noisy binary search and its applications. In *SODA*, 881–890.

Khintchine, A. 1923. Über dyadische Brüche. *Math Z.* 18:109–116.

Kleinberg, R. 2005. *Online Decision Problems with Large Strategy Sets*. Ph. D. thesis, MIT, Boston, MA.

Kleinberg, R. Spring 2007a. Lecture notes: *CS683: Learning, Games, and Electronic Markets* (week 8). Available at `http://www.cs.cornell.edu/courses/cs683/2007sp/lecnotes/week8.pdf`.

Kleinberg, R. Spring 2007b. Lecture notes: *CS683: Learning, Games, and Electronic Markets* (week 9). Available at `http://www.cs.cornell.edu/courses/cs683/2007sp/lecnotes/week9.pdf`.

Kleinberg, R., A. Niculescu-Mizil, and Y. Sharma. 2010. Regret bounds for sleeping experts and bandits. *Machine Learning Journal*. Preliminary version appeared in Conference on Learning Theory (COLT) 2008, pages 425–436.

Kleinberg, R., A. Slivkins, and E. Upfal. 2008. Multi-Armed Bandits in Metric Spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, 681–690.

Lahaie, S., D. M. Pennock, A. Saberi, and R. V. Vohra. 2007. *In n. nisan, t. roughgarden, e. tardos, and v. vazirani (eds.) chapter 28, sponsored search auctions.* Cambridge University Press.

Lai, T., and H. Robbins. 1985a. Asymptotically efficient Adaptive Allocation Rules. *Advances in Applied Mathematics* 6:4–22.

Lai, T. L., and H. Robbins. 1985b. Asymptotically efficient adaptive allocations rules. *Adv. in Appl. Math.* 6:4–22.

Langford, J., and T. Zhang. 2007. The epoch-greedy algorithm for multiarmed bandits with side information. In *NIPS*.

Lavi, R., A. Mu'alem, and N. Nisan. 2003. Towards a characterization of truthful combinatorial auctions. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 574.

Lavi, R., and N. Nisan. 2005. Online ascending auctions for gradually expiring items. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 1146–1155.

Littlestone, N., and M. K. Warmuth. 1994. The weighted majority algorithm. *Inf. Comput.* 108 (2): 212–261. An extended abstract appeared in IEEE Symposium on Foundations of Computer Science, 1989, pp. 256–261.

Madani, O., and D. DeCoste. 2005. Contextual recommender problems. In *UBDM*.

Mehta, A., A. Saberi, U. Vazirani, and V. Vazirani. 2007. Adwords and generalized online matching. *J. ACM* 54 (5): 22.

Myerson, R. B. 1981. Optimal Auction Design. *Mathematics of Operations Research* 6:58–73.

Nazerzadeh, H., A. Saberi, and R. Vohra. 2008. Dynamic cost-per-action mechanisms and applications to online advertising. In *17th Intl. World Wide Web Conf. (WWW)*, 179–188.

Nisan, N., and A. Ronen. 2001. Algorithmic Mechanism Design. *Games and Economic Behavior* 35 (1-2): 166–196.

Nisan, N., T. Roughgarden, E. Tardos, and V. V. (eds.). 2007. *Algorithmic game theory*. Cambridge University Press.

Papadimitriou, C., M. Schapira, and Y. Singer. 2008. On the hardness of being truthful. In *IEEE Symp. on Foundations of Computer Science (FOCS)*.

Radlinski, F., R. Kleinberg, and T. Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *ICML*, 784–791.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* 58:527–535.

Roughgarden, T. 2008. An algorithmic game theory primer. IFIP International Conference on Theoretical Computer Science (TCS). An invited survey.

Varian, H. R. 2007, December. Position auctions. *International Journal of Industrial Organization* 25 (6): 1163–1178.

Vovk, V. G. 1990. Aggregating strategies. In *COLT*, 371–386.

Vovk, V. G. 1998. A game of prediction with expert advice. *J. Comput. Syst. Sci.* 56 (2): 153–173. An extended abstract appeard in COLT 1995, pp. 51–60.